



**Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia de Software**

**USO DO *SCRUM* EM UM PROCESSO DE  
GESTÃO DE DEMANDAS DE  
DESENVOLVIMENTO DE SOFTWARE POR  
TERCEIROS PARA UM ÓRGÃO PÚBLICO  
FEDERAL BRASILEIRO**

**Autor: Luiz Pereira de Souza Sobrinho**

**Orientadora: Dra. Rejane M da Costa Figueiredo**

**Brasília, DF  
2014**



**LUIZ PEREIRA DE SOUZA SOBRINHO**

**USO DO *SCRUM* EM UM PROCESSO DE GESTÃO DE DEMANDAS DE  
DESENVOLVIMENTO DE SOFTWARE POR TERCEIROS PARA UM  
ÓRGÃO PÚBLICO FEDERAL BRASILEIRO**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Rejane Maria da Costa Figueiredo

**Brasília, DF  
2014**



**USO DO *SCRUM* EM UM PROCESSO DE GESTÃO DE DEMANDAS DE  
DESENVOLVIMENTO DE SOFTWARE POR TERCEIROS PARA UM  
ÓRGÃO PÚBLICO FEDERAL BRASILEIRO**

**Luiz Pereira de Souza Sobrinho**

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software da Faculdade UnB Gama - FGA, da Universidade de Brasília, em 25/06/2014, apresentada e aprovada pela banca examinadora abaixo assinada:

---

**Dra. Rejane Maria da Costa Figueiredo, UnB/ FGA**  
Orientadora

---

**Prof. Dr. Luiz Carlos Miyadaira, UnB/ FGA**  
Membro Convidado

---

**Prof. Msc.: George Marsicano Corrêa, UnB/ FGA**  
Membro Convidado

Brasília, DF  
2014

## **AGRADECIMENTOS**

A Deus, Pai, Filho e Espírito Santo pelo dom da vida e por todas as misericórdias a mim concedidas. Sem Ti Senhor nada posso.

Agradeço à minha mãe, Maria Lucineide, pelo apoio e amor incondicional e por me dar a crença de que a educação é o único caminho.

Aos meus padrinhos de batismo, Luiz e Marta e aos meus tios Deoclécio e Zilda, que me apoiaram e cuidaram para que eu pudesse estudar.

A todos os meus familiares que me concederam suas casas e me aceitaram em suas famílias, sempre com o carinho e atenção muito maiores do eu esperava, em especial aos meus tios Nelson e Heloisa, Rainel e Maria da Paz e Jaime e Carmem. Obrigado, os vossos carinhos me sustentaram até aqui.

Agradeço aos meus amigos e familiares, que são muitos para que os liste aqui.

À minha orientadora, Prof.<sup>a</sup> Dr.<sup>a</sup> Rejane, pelas orientações sempre oportunas, cresci e aprendi mais com você do que posso dizer. Agradeço pelo tempo concedido pela paciência, gentileza, cordialidade e confiança a mim doados, não só como orientadora deste trabalho mas ao longo de todo curso.

À Prof.<sup>a</sup> Msc. Fabiana Freitas, pela paixão e entusiasmo que passou ao me ensinar e orientar. Aos professores da frente de pesquisa de Melhoria de Processos, Prof. Msc. Ricardo Ajax e Prof.<sup>a</sup> Msc. Elaine Venson, pelas orientações, confiança e amizade a mim dedicadas neste tempo de projeto.

Obrigado.

*“Tudo que o homem não conhece não existe para ele. Por isso o mundo tem para cada um, o tamanho que abrange o seu conhecimento”*

*(Carlos Bernardo Gozález Pecotche)*

# SUMÁRIO

<b>AGRADECIMENTOS .....</b>	<b>IV</b>
<b>LISTA DE FIGURAS .....</b>	<b>VIII</b>
<b>LISTA DE TABELAS .....</b>	<b>IX</b>
<b>RESUMO .....</b>	<b>X</b>
<b>CAPÍTULO 1 - INTRODUÇÃO .....</b>	<b>11</b>
1.1. Contexto.....	12
1.2. Questão de Pesquisa.....	13
1.3. Objetivos.....	13
1.4. Metodologia de Pesquisa .....	14
1.5. Organização do Trabalho.....	16
<b>CAPÍTULO 2 – CONTRATAÇÕES DE SERVIÇOS DE TECNOLOGIA DA INFORMAÇÃO POR ORGANIZAÇÕES PÚBLICAS BRASILEIRAS .....</b>	<b>18</b>
2.1. Considerações Iniciais do Capítulo .....	19
2.2. Contratação de Serviços de Tecnologia da Informação pelo Setor Público Brasileiro .....	19
2.2.1.       Processos de Contratação de Software para a Administração Pública Federal       22	
2.2.2.       Gerenciamento do Contrato .....	25
2.3. Considerações Finais do Capítulo .....	27
<b>CAPÍTULO 3 - <i>FRAMEWORK SCRUM</i> .....</b>	<b>28</b>
3.1. Considerações Iniciais do Capítulo .....	29
3.2. Criação do <i>Scrum</i> .....	29
3.2.1.       O Processo do <i>Framework Scrum</i> .....	30
3.3. Empregos do <i>Scrum</i> .....	35
3.3.1.       O <i>Scrum</i> no Governo.....	35
3.3.2.       Estratégia de emprego do <i>Scrum</i> .....	40
3.4. O <i>Scrum</i> e a Transferência de Conhecimento .....	56
3.4.1.       Ciclo SECI no <i>Scrum</i> para Governo .....	56

3.4.2. Visão Sistêmica e Aprendizagem Organizacional .....	57
3.5. Considerações finais do Capítulo .....	59
<b>CAPÍTULO 4 - MATERIAIS E MÉTODOS .....</b>	<b>61</b>
4.1. Considerações Iniciais do Capítulo .....	62
4.2. Caracterização do Órgão.....	62
4.3. Gestão de Projeto do Órgão.....	67
4.4. Considerações Finais do Capítulo .....	71
<b>CAPÍTULO 5 - ADAPTAÇÕES DO <i>FRAMEWORK SCRUM</i> PARA UM ÓRGÃO PÚBLICO BRASILEIRO .....</b>	<b>73</b>
5.1. Considerações Iniciais do Capítulo .....	74
5.2. Processo Ágil de Gestão de Demandas de Desenvolvimento de Software - PAGDDS.....	74
5.3. O Time <i>Scrum</i> e o POP .....	80
5.4. Os eventos <i>Scrum</i> no PAGDDS .....	81
5.5. Os artefatos <i>Scrum</i> no PAGDDS .....	84
<b>CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>87</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>90</b>
<b>APÊNDICE – APRESENTAÇÃO DOS SUBPROCESSOS DO PAGDDS.....</b>	<b>100</b>

# LISTA DE FIGURAS

FIGURA 1 – METODOLOGIA DA PESQUISA.....	15
FIGURA 2 – CONTEXTO DE ELABORAÇÃO DO PLANEJAMENTO DE TI (BRASIL, 2011A).....	24
FIGURA 3 – CONTEXTO DO PLANEJAMENTO DAS CONTRATAÇÕES DE SOLUÇÕES DE TI (BRASIL, 2012C).....	25
FIGURA 4 – GESTÃO DO CONTRATO DE TI (BRASIL, 2011A).....	26
FIGURA 5 – <i>FRAMEWORK SCRUM</i> BASEADO NO GUIA <i>SCRUM</i> (SCHWABER; SUTHERLAND, 2013).....	31
FIGURA 6 – COMPOSIÇÃO DO TIME <i>SCRUM</i> BASEADO NO GUIA <i>SCRUM</i> (SCHWABER; SUTHERLAND, 2013).....	32
FIGURA 7 – ESTRATÉGIAS DE EXECUÇÃO DAS <i>SPRINTS</i> ADAPTADO DE (SUTHERLAND, 2005) (TAKEUCHI; NONAKA, 1986).....	41
FIGURA 8 – <i>FRAMEWORK</i> DE CICLOS DE CONTROLE COM <i>SCRUM</i> ADAPTADO DE (VANHANEN; ITKONEN; SULONEN, 2003) (RAUTIAINEN; VUORNOS; LASSENIUS, 2003A).....	42
FIGURA 9 – MODELO <i>STAGE-GATE</i> ADAPTADO (COOPER, 1990).....	43
FIGURA 10 – FLUXO DE TRABALHO DO <i>SCRUM</i> ADAPTADO DE (JAKOBSEN E SUTHERLAND, 2009).....	50
FIGURA 11 – PROCESSO DE SOFTWARE ÁGIL ADAPTADO DE (COHN; SIM; LEE, 2009).....	51
FIGURA 12 – ESTRUTURA DO <i>SCALED AGILE FRAMEWORK</i> ADAPTADO DE (LEFFINGWELL, 2013B).....	53
FIGURA 13 - ESPIRAL SECI ADAPTADO DE NONAKA & TAKEUCHI 1997 (BRITO, 2013).....	57
FIGURA 14 – CONTEXTO CONTRATUAL DO ESTUDO DE CASO PARA OS ENVOLVIDOS NO DESENVOLVIMENTO DE SOFTWARE.....	65
FIGURA 15 – ORGANOGRAMA CGTI DO ESTUDO DE CASO.....	66
FIGURA 16 – COMPOSIÇÃO DO CGP (FONTE: AUTOR).....	68
FIGURA 17 – FLUXO DA MGPTI (REFERÊNCIA).....	69
FIGURA 18 – MACROPROCESSO PAGDDS.....	76
FIGURA 19 – <i>FRAMEWORK SCRUM</i> ADAPTADO PARA O ESTUDO DE CASO.....	79
FIGURA 20 – EVENTOS <i>SCRUM</i> SUBSTITUÍDOS PARA O ESTUDO DE CASO.....	82
FIGURA 21 – INSERÇÃO DA REUNIÃO DE RETROSPECTIVA DA <i>SPRINT</i> NA REUNIÃO DE REVISÃO DA <i>SPRINT</i> .....	84
FIGURA 22 – ALTERAÇÃO DOS ARTEFATOS <i>SCRUM</i> PARA O ESTUDO DE CASO.....	86
FIGURA 23 – SUBPROCESSO DE PRÉ-PROJETO DO PAGDDS.....	101
FIGURA 24 – SUBPROCESSO DE REVISÃO DO PLANEJAMENTO DO PROJETO.....	102
FIGURA 25 – SUBPROCESSO DE PLANEJAMENTO DA <i>RELEASE</i> NO PAGDDS.....	103
FIGURA 26 – SUBPROCESSO DE EXECUÇÃO NO PAGDDS.....	104
FIGURA 27 - SUBPROCESSO DE HOMOLOGAÇÃO DO PAGDDS.....	106
FIGURA 28 - SUBPROCESSO DE IMPLANTAÇÃO DO PAGDDS.....	107



## LISTA DE TABELAS

QUADRO 1 – CARACTERIZAÇÃO DOS RELATÓRIOS SOBRE METODOLOGIAS ÁGEIS NO GOVERNO (AUTOR).....	37
QUADRO 2 – PRÁTICAS RECOMENDADAS EM CASOS GOVERNAMENTAIS (AUTOR).....	39
QUADRO 3 – <i>SCRUM</i> E PMBOK ADAPTADO DE (SCHWALBE, 2012).....	45
QUADRO 4 – OS MELHORES PADRÕES <i>SCRUM</i> ADAPTADO DE (SUTHERLAND; HARRISON; RIDDLE, 2014).....	46
QUADRO 5 – DIFERENÇAS ENTRE <i>SCRUM</i> DE DESENVOLVIMENTO E DE GERENCIAMENTO DE PRODUTO ÁGIL ADAPTADO DE (VLAANDEREN ET AL., 2011).....	48
QUADRO 6 – <i>SCRUM</i> E AS DISCIPLINAS PARA APRENDIZAGEM ORGANIZACIONAL (AUTOR).....	58
QUADRO 7 – OBJETIVOS DAS DECISÕES.....	70
QUADRO 8 – COMPARAÇÃO ENTRE PAPÉIS DA MGPTI E <i>SCRUM</i> .....	71
QUADRO 9 – <i>TIME-BOX</i> DOS SUBPROCESSOS DO PAGDDS.....	77
QUADRO 10 – ESTRUTURA DO SAFE E ESTUDO DE CASO.....	78

## RESUMO

O governo federal brasileiro é um grande consumidor do serviço de desenvolvimento software através de contratação, que necessita de insumos para auditoria e gestão contratual o que induz a um processo de desenvolvimento de software dirigido a planos baseados no modelo Cascata, tanto no Brasil como em outros países do mundo. Porém com o aumento da popularidade das metodologias ágeis de desenvolvimento de software tem-se encontrado iniciativas de adequação destas ao ambiente contratual, inclusive no Brasil, como alternativa ao desenvolvimento cascata ou baseado no RUP. Reconhecendo o Scrum como a mais emblemática representante das metodologias ágeis, o complexo contexto legal para contratações no Brasil e dados os relatos acadêmicos de adaptação e combinação desta metodologia com outras, este trabalho utiliza um estudo de caso de um órgão público federal brasileiro para, dadas as características deste órgão, propor as adaptações necessárias ao contexto de contratação no setor público federal brasileiro. Dentre as adaptações definidas destacam-se os eventos voltados ao *Product Owner*, a proposta do papel do *Product Owner Partner* (POP), além de outras adaptações em artefatos mantendo-se o foco do desenvolvimento no produto de software com o mínimo de documentação exigida.

**Palavras-chave:** *Scrum*, Adaptação, Governo, Contratação, Terceirização, Administração Pública Federal, Desenvolvimento de Software, Estudo de Caso.

---

## **CAPÍTULO 1 - INTRODUÇÃO**

## 1.1. CONTEXTO

A Engenharia de Software é a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção do software (ISO/IEC/IEEE, 2010). Neste esforço, várias metodologias de desenvolvimento de software foram propostas ao longo dos anos.

O Manifesto para o desenvolvimento ágil de software (BECK et al., 2001) se caracteriza como um marco de uma nova geração de metodologias (DINGSØYR et al., 2012). Esse manifesto conta com o endosso de dezessete profissionais da engenharia de software, vários deles criadores e mantenedores de metodologias ágeis (JANES; SUCCI, 2012).

Muitas discussões tem sido realizadas entre o mérito das metodologias ágeis e de metodologias dirigidas a planos (DINGSØYR et al., 2012). As metodologias dirigidas a planos são uma abordagem mais adequada ao desenvolvimento de sistemas críticos, com poucas mudanças e requisitos estáveis, enquanto as metodologias ágeis de desenvolvimento de software são melhor aplicadas a situação oposta, permitindo maior flexibilidade do projeto e dos requisitos e exigindo maior participação do cliente (KUMAR; BHATIA, 2014).

Um dos principais desafios nas pesquisas em metodologias ágeis é a sua adequação ao ambiente contratual, baseado mais em planos e processos do que interações e pessoas (MAURER; MELNIK, 2007) que requerem adaptações e balanceamento entre agilidade e disciplina (BOEHM; TURNER, 2003).

As metodologias ágeis mais pesquisadas são o *Scrum*, voltado ao gerenciamento e o *Extreme Programming* (XP), voltado ao desenvolvimento (DINGSØYR; DYBÅ; ABRAHAMSSON, 2008). No Brasil, o *Scrum* é a metodologia ágil mais utilizada, sendo seguida por sua combinação com o XP (MELO et al., 2013), embora ainda sejam necessárias pesquisas a respeito do uso do *Scrum* (HUMMEL, 2014).

Pesquisas empíricas indicaram que o uso de metodologias ágeis, incluindo o *Scrum*, tem proporcionado melhoria na qualidade do produto de software, com aumento da taxa de entrega e maior satisfação dos clientes (DOWNEY; SUTHERLAND, 2013)(WILLIAMS et al., 2011)(MUNDRA; MISRA; DHAWALE, 2013).

Com a intenção de obter esses benefícios, vários órgãos públicos federais brasileiros têm buscado adotar metodologias ágeis de desenvolvimento de *software* em seus contratos

(BRASIL, 2013), que devem respeitar um compêndio vasto e complexo de legislações (CRUZ, 2008). Outros países também têm realizado estudos e publicações sobre o uso de metodologias ágeis em seus governos (EUA, 2012) (INGLATERRA, 2012).

Seguindo a tendência de adoção de ágeis, o Centro de Qualidade e Testes de Software (CQTS), da Universidade de Brasília (UnB) estabeleceu um Termo de Cooperação Técnica com um Ministério do Governo Federal, com um projeto de pesquisa e desenvolvimento, intitulado *Framework* de Soluções de TI.

Esse projeto conta com três frentes de trabalho relacionadas as áreas de melhoria priorizadas pelo órgão. Entre essas se encontra a frente de Melhoria de Processos de Software, da qual este autor faz parte desde o início do projeto, dezembro de 2012, e que tem o objetivo de propor um Processo de Gestão de Demandas de Desenvolvimento de Software por Terceiros segundo os princípios e práticas ágeis.

Neste contexto Brito (2013) propôs as atividades de transferência de conhecimento para este processo, sendo encargo deste trabalho apresentar o uso do *Scrum* em uma versão atualizada deste processo.

## 1.2. QUESTÃO DE PESQUISA

Dada a tendência de uso de metodologias ágeis no governo (EUA, 2012) (INGLATERRA, 2012) (BRASIL, 2013), e de sua dificuldade de adaptação em um contexto rígido e de terceirização (BATRA, 2009), somados a um complexo conjunto de legislações (CRUZ, 2008) e da necessidade de mais pesquisas sobre o uso do *Scrum* (HUMMEL, 2014), e sendo o *Scrum*, um *framework* para apoio a gestão, a questão de pesquisa deste trabalho é:

***Como empregar o Framework Scrum na definição de processo de gestão de demandas de desenvolvimento de software por terceiros para um órgão do governo público federal brasileiro?***

## 1.3. OBJETIVOS

O objetivo deste trabalho foi apoiar a definição de um processo de gestão de demandas de desenvolvimento de software por terceiros para um órgão do governo federal brasileiro, utilizando o *Framework Scrum*.

Os objetivos específicos definidos foram:

1. Identificar e analisar diferentes empregos do *Framework Scrum* por organizações;
2. Selecionar e caracterizar um órgão do governo federal brasileiro como objeto de estudo de caso;
3. Identificar e adequar práticas para a definição do processo de gestão de demandas de desenvolvimento de software por terceiros para o órgão do governo federal brasileiro selecionado.

## 1.4. METODOLOGIA DE PESQUISA

Segundo a classificação de Gil (2008), esta pesquisa é definida como: pesquisa de natureza aplicada e abordagem qualitativa, pois trata-se de um estudo aplicado de um *Framework* para um órgão público, cuja a análise dos dados é qualitativa, não envolvendo análises quantitativas.

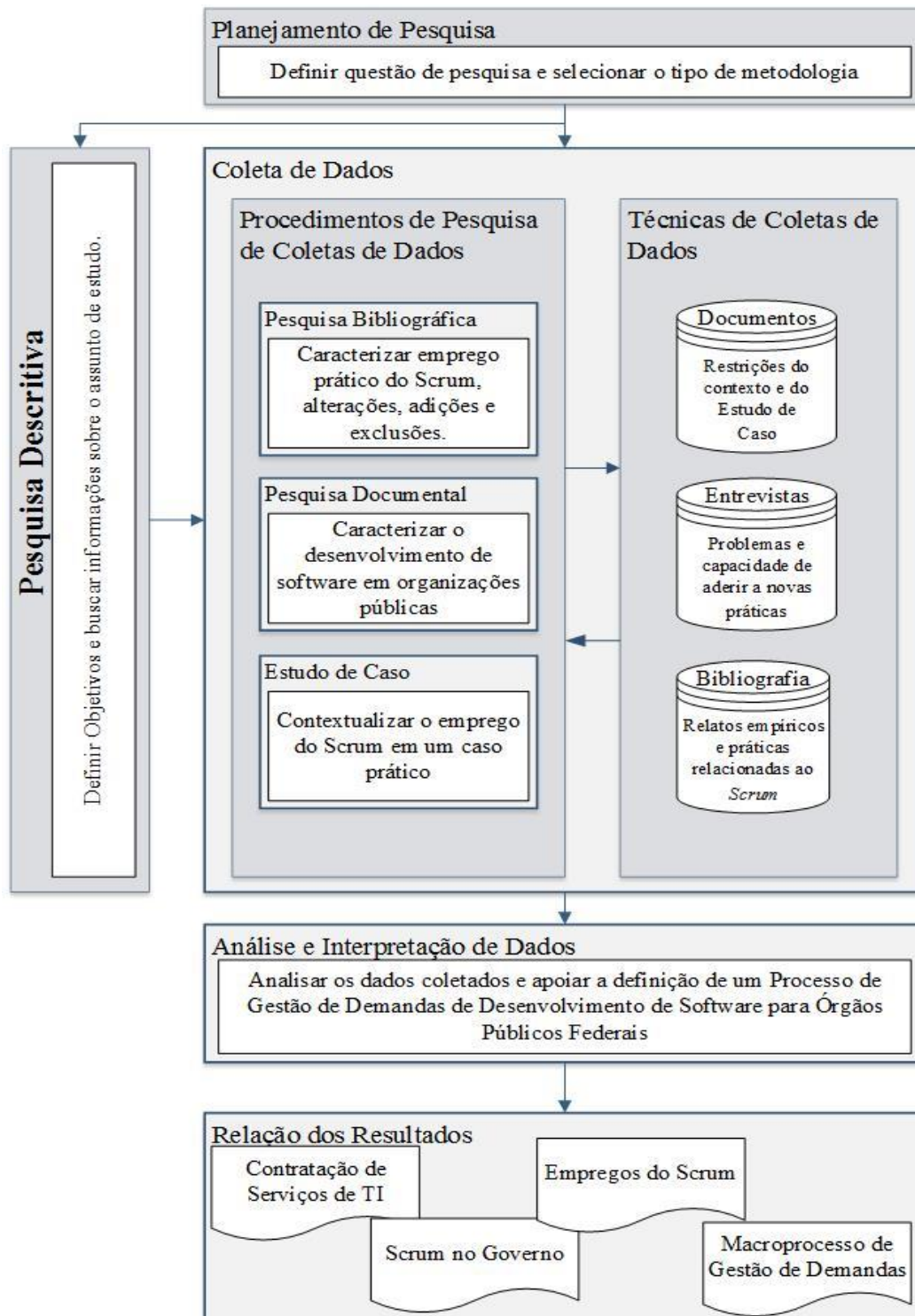
Quanto ao tipo é classificada como descritiva, pois descreve um estudo de adaptação do *Framework Scrum* a partir da revisão bibliográfica e da legislação brasileira. Quanto aos procedimentos técnicos, bibliográfica e documental, seguida do emprego de estudo de caso, com a seleção de um órgão público federal brasileiro.

Na Figura 1 apresenta-se a organização das fases adotadas nesta pesquisa. Segundo André (2005): Fase Planejamento, Coleta de Dados e Análise de Dados.

- **Planejamento:** em que se definem o caso de estudo e suas questões, estabelecendo os contatos, os procedimentos e os instrumentos de coleta de dados;
- **Coleta dos dados:** em que se aplicam os procedimentos de coleta;
- **Análise dos dados:** em que se realiza a análise dos dados e a redação dos resultados.

A Fase de Coleta de Dados apresenta as técnicas empregadas para cada um dos objetivos específicos definidos neste trabalho. As técnicas utilizadas foram:

- **Revisão Bibliográfica:** pesquisa realizada nas principais bases de dados de conferências e revistas em engenharia de software, buscando a caracterização das contratações de serviços de TI por órgãos públicos brasileiros e internacionais e reconhecer adaptações do *Framework Scrum*;



**Figura 1 – Metodologia da Pesquisa**

- **Análise Documental:** utilizada tanto para caracterização da aplicação do *Scrum* no setor público, a partir de relatórios nacionais e internacionais, quanto para caracterização do órgão estudado, seus processos vigentes, seu contexto atual, com anseios e limitações para adoção de metodologias ágeis;
- **Entrevistas não estruturadas:** foram utilizadas para a caracterização do órgão segundo seu estado e processos atuais e sua expectativa com a adoção de metodologias ágeis.

A fase de análise representa tanto a definição de proposta de emprego do *Framework Scrum* para o órgão quanto a validação destas propostas por representantes do órgão, caracterizando assim, ciclos iterativos de definição e ajustes, como recomendado por Gil (2008) para pesquisas de cunho qualitativo que não possuem modelo teórico de análise definido.

## 1.5. ORGANIZAÇÃO DO TRABALHO

Este trabalho possui 06 capítulos. Neste Capítulo de Introdução foram apresentados: o contexto do trabalho; o problema; os objetivos e a metodologia de pesquisa adotada; seguida da organização do trabalho.

No Capítulo 2 – *Contratação de Serviços de tecnologia da informação*, estabelecem-se as características do relacionamento cliente-fornecedor, a partir dos princípios da administração pública federal brasileira.

No Capítulo 3 – *Framework Scrum*, apresentam-se os conceitos do *Framework Scrum*, com breve histórico, o relacionamento com transferência de conhecimento e aprendizagem organizacional, alguns estudos relacionados ao emprego do *Scrum* e algumas adaptações e combinações.

No Capítulo 4 – *Materiais e Métodos*, a partir da breve descrição da metodologia empregada neste trabalho, e sintetizada na Figura 1 – Metodologia da Pesquisa, este capítulo apresenta os resultados da análise documental e de entrevistas para a caracterização do Estudo de caso.

No Capítulo 5 – *Adaptações do Scrum para o Caso Estudado*, apresenta-se o processo de gestão de demandas definido pelo grupo de pesquisa no contexto do termo de cooperação e as adaptações do *Scrum* associadas a ele.



No Capítulo 6 – *Conclusões e Trabalhos futuros*, apresentam-se as considerações finais deste trabalho e possíveis trabalhos futuros.

---

## **CAPÍTULO 2 – CONTRATAÇÕES DE SERVIÇOS DE TECNOLOGIA DA INFORMAÇÃO POR ORGANIZAÇÕES PÚBLICAS BRASILEIRAS**

## 2.1. CONSIDERAÇÕES INICIAIS DO CAPÍTULO

Este Capítulo caracteriza o contexto de contratações de serviços de tecnologia da informação pelas organizações públicas brasileiras, a partir da legislação vigente. O foco é a contratação de fábricas de software, tanto do ponto de vista da legislação quanto de processos que possam apoiar a contratação assim como limites e restrições na condução deste contrato em organizações públicas federais brasileiras.

## 2.2. CONTRATAÇÃO DE SERVIÇOS DE TECNOLOGIA DA INFORMAÇÃO PELO SETOR PÚBLICO BRASILEIRO

O Decreto-Lei nº 200, de 25 de fevereiro de 1967 (BRASIL, 1967), estabelece cinco princípios fundamentais da administração pública que são:

- **Planejamento:** define a obrigatoriedade do planejamento, com definição de objetivos, recursos financeiros e prazos, de forma a serem acompanhados;
- **Coordenação:** reafirma a hierarquia propondo a criação de coordenações descentralizadas para cada setor administrativo;
- **Descentralização** estabelece o uso amplo da descentralização em três níveis, sendo um deles a contratação sempre que possível e haja competência no mercado para tanto;
- **Delegação de Competência:** prevê a delegação como instrumento de descentralização de forma a assegurar rapidez e objetividade das decisões por pessoas próximas aos problemas e situações a serem solucionadas. Porém esta delegação deve ser formalizada indicando com precisão a autoridade delegante, a autoridade delegada e as atribuições do objeto de delegação, ou seja, quem tem o poder, para quem está passando e qual o poder passar a respeito do que;
- **Controle:** deve ser exercido em três níveis: pela chefia, por órgãos próprios de cada sistema e pelos órgãos do sistema de contabilidade e auditoria (o Tribunal de Contas da União é um destes representantes no âmbito federal).

O Decreto nº 2.271, de 7 de julho de 1997 (BRASIL, 1997) recomenda que todos os produtos ou serviços que não tiverem ligação direta com a finalidade da instituição sejam terceirizados, incluindo entre estes os bens e serviços de informática.

As contratações de serviços e produtos no governo brasileiro devem dar prioridade a licitação garantindo assim o princípio constitucional da isonomia<sup>1</sup> (BRASIL, 1993). A licitação é o processo formal utilizado pela Administração Pública para publicar, através de edital, sua necessidade de bens e serviços e selecionar a proposta mais vantajosa (BRASIL, 2010a).

A Lei nº 8.666, de 21 de junho de 1993 (BRASIL, 1993) institui normas para a licitação e contratos da Administração Pública. E define ainda em seu Artigo 73 que o recebimento dos serviços deverá ser efetuado em duas etapas: provisório, até 15 dias após o comunicado de conclusão; e definitivo, após verificação de adequação do serviço aos termos contratuais. Estabelece ainda nos artigos 86 e 87, sanções administrativas contra a contratada por atraso injustificado de execução do contrato ou por inexecução total ou parcial do contrato, garantido em ambos os casos pleno direito de defesa.

O Tribunal de Contas da União e o Senado Federal publicaram em conjunto o livro *Licitações & Contratos* (BRASIL, 2010a) que dentre outros conceitos define os nove princípios do processo licitatório, como apresentado a seguir:

- **Princípio da Legalidade:** obediência as normas e princípios em vigor;
- **Princípio da Isonomia:** tratamento igual a todos os interessados. Garante a competição;
- **Princípio da Impessoalidade:** tomar decisões a partir de critérios objetivos e previamente estabelecidos, afastando discricionariedade e o subjetivismo;
- **Princípio da Moralidade e da Probidade Administrativa:** além de lícita a conduta dos agentes públicos deve ser compatível com a moral, ética;

---

<sup>1</sup> Isonomia: Princípio, assegurado pela Constituição, segundo o qual todos são iguais perante a lei, não podendo haver nenhuma distinção em relação a pessoas que estejam na mesma situação (IDICIONÁRIO, 2013)

- **Princípio da Publicidade:** todas as informações a respeito do processo devem estar disponíveis a qualquer interessado, isto se dá por meio da publicação do Edital;
- **Princípio da Vinculação ao Instrumento Convocatório:** nada pode ser criado ou feito sem que haja previsão no instrumento convocatório;
- **Princípio do Julgamento Objetivo:** critérios objetivos definidos no ato convocatório para julgar as propostas, impede o subjetivismo na seleção do fornecedor;
- **Princípio da Celeridade:** não interpor procedimentos e rigorisismos excessivos e formalidades desnecessárias para a tomada de decisão, que deve ser, sempre que possível, tomadas no momento da sessão, no caso da modalidade pregão; e
- **Princípio da Competição:** buscar o maior número de competidores não definindo, no edital ou seus anexos, restrições excessivas que diminuam ou vedem a possibilidade de competição.

A licitação possui quatro modalidades: concorrência, tomada de preços, convite e pregão (BRASIL, 2010a). O pregão é a forma obrigatória para a contratação de bens e serviços comuns, como definido na Lei nº 10.520 de julho de 2002 (BRASIL, 2002) que o regulamenta.

Segundo o Acórdão TCU 1.287/2008 - Plenário (BRASIL, 2008), bem ou serviço comum é aquele que pode ter seus padrões de desempenho e qualidade objetivamente definidos pelo edital, por meio de especificações usuais no mercado.

A qualificação de um bem ou serviço como comum fica a cargo do gestor responsável pela licitação. O Acórdão TCU 188/2010 Plenário (BRASIL, 2010b) estabelece que a complexidade do serviço não exclui sua qualificação como comum, desde que haja especificações usuais no mercado e sejam definidos padrões objetivos no edital.

No livro *Licitações & Contratos* (BRASIL, 2010a) estabelece-se que a escolha de bens e serviços comuns deve ser feita com base somente nos preços ofertados, por serem comparáveis entre si e não necessitarem de avaliação minuciosa. Define ainda que os Editais

para bens e serviços comuns, no qual pode se enquadrar o desenvolvimento e manutenção de sistemas de informação, deve possuir um Termo de Referência.

O Termo de Referência determina o escopo do contrato, seu custo e prazo, assim como os critérios de aceitação e avaliação dos custos, sanções por inadimplência e os procedimentos de fiscalização e gerenciamento do contrato (BRASIL, 2010a). Sendo pertinente define-se o prazo de garantia para o produto ou serviço objeto do edital.

Nota-se que a legislação para a contratação é extensa, auto complementar e complexa. Cruz (2008) definiu um Quadro-Referencial Normativo para contratações de Tecnologia da Informação na Administração Pública Federal Brasileira, dado um ciclo de vida básico de contratações, para cada fase e atividades foi associada a legislação pertinente.

### **2.2.1. PROCESSOS DE CONTRATAÇÃO DE SOFTWARE PARA A ADMINISTRAÇÃO PÚBLICA FEDERAL**

O Decreto-Lei nº 200, de 25 de fevereiro de 1967 (BRASIL, 1967) define entre os princípios da administração pública o controle que deve ser exercido em três instâncias: pela chefia do respectivo órgão, seção ou coordenação; pelo órgão próprio de cada sistema; e pelos órgãos do sistema de contabilidade e auditoria.

A chefia do respectivo órgão é a responsável pela licitação e definição do Termo de Referência, e é responsável por gerir e fiscalizar o contrato (BRASIL, 1993).

A respeito dos serviços de TI o sistema responsável no âmbito federal é o Sistema de Administração dos Recursos de Tecnologia da Informação (SISP). Este sistema foi instituído em 1994 e atualizado em 2011, para organizar a operação, controle, supervisão e coordenação dos recursos de informação e informática dos órgãos públicos federais e está sujeito ao Ministério do Planejamento, Orçamento e Gestão (MP ou MPOG) (BRASIL, 2012a).

O seu principal órgão é a Secretaria de Logística e Tecnologia da Informação (SLTI) do Ministério do Planejamento Orçamento e Gestão (BRASIL, 2012b) que tem como um de seus objetivos normatizar, promover e coordenar ações junto aos órgãos do SISP quanto a gestão e governança de tecnologia da informação; gestão de pessoas e capacitação; e melhoria de processos de desenvolvimento de sistemas. (BRASIL, 2012a).

O controle contábil é realizado de forma externa pelo Tribunal de Contas da União (TCU) e sua Secretaria de Fiscalização em Tecnologia da Informação (SEFTI) que auditam

as contratações da Administração Pública Federal. Em sua página na internet sobre contratação de tecnologia da informação (BRASIL, 2014) o TCU apresenta acórdãos, livros e revistas publicados a respeito deste tema.

No uso de suas atribuições a SLTI publicou, originalmente em 2008 e atualizou em 2010, a Instrução Normativa MP/SLTI N°04 (IN4) que dispõe sobre o processo de contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal (BRASIL, 2010c).

A IN4 (BRASIL, 2010c) define três fases do processo de contratação: Planejamento da Contratação, Seleção do Fornecedor e Gerenciamento do Contrato, além de definir papéis e artefatos para cada uma destas fases.

Ambos os órgãos de controle (SLTI e TCU) tem adotado a prática de prover orientação através de guias relacionados a Instrução Normativa MP/SLTI N°04 (IN04). O TCU lançou o Guia de boas práticas em contratação de soluções de tecnologia da informação: riscos e controles para o planejamento da contratação (BRASIL, 2012c) e a SLTI lançou Guia de Boas Práticas em Contratação de Soluções de TI que apresenta o Modelo de Contratação de TI (MCTI) (BRASIL, 2011a)

O Guia do TCU apresenta uma série de riscos relacionados a contratação de TI e suas mitigações, dentre as quais sugere que todas as relações entre contratada e contratante sejam documentadas, de modo a permitir a adequada auditoria, tendo em vista que esta pode ocorrer apenas anos depois (BRASIL, 2012c).

No sentido de orientar a contratação e execução do contrato têm-se ainda o Processo de Contratação de Serviços de TI (PCSTI)<sup>2</sup> (CRUZ; ANDRADE; FIGUEIREDO, 2011), que define o mesmo processo em termos de atores, atividades e tarefas, apresentando inclusive *templates* de seus artefatos propostos.

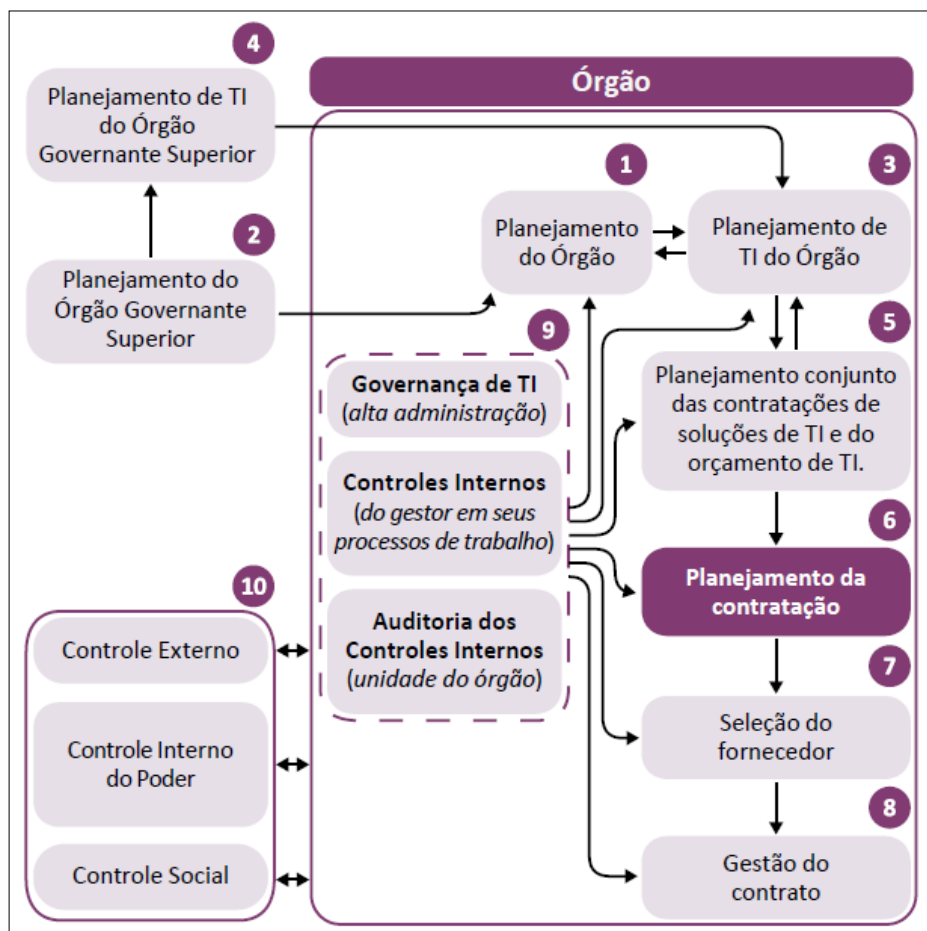
O planejamento da contratação deve ocorrer em concordância com outros planejamentos existentes (BRASIL, 2010c). A Figura 2, retirada do Guia da SLTI, apresenta quatro planos: o PPA que é o Plano Plurianual definido para todo o poder executivo; em

---

<sup>2</sup> Projeto vencedor do PBQP Software de 2011 - Programa Brasileiro da Qualidade e Produtividade em Software foi criado em 1993, com apoio da SEPIN/MCT – Secretaria de Política de Informática, do Ministério da Ciência e Tecnologia.







**Figura 3 – Contexto do planejamento das contratações de soluções de TI (BRASIL, 2012c)**

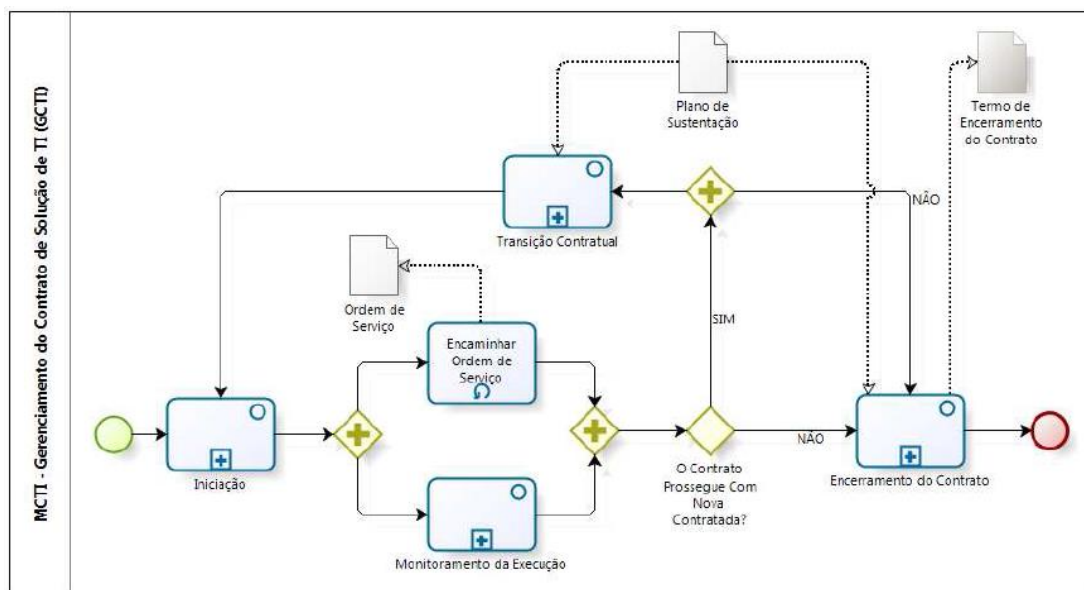
### 2.2.2. GERENCIAMENTO DO CONTRATO

A mesma IN04 (BRASIL, 2010c) proíbe a contratação de gestão de processos de TI, que deve ficar a cargo do órgão e seus funcionários.

A fase de gerenciamento do Contrato tem o objetivo de:

*“...acompanhar e garantir a adequada prestação dos serviços e o fornecimento dos bens que compõem a Solução de Tecnologia da Informação durante todo o período de execução do contrato” (BRASIL, 2010c).*

Esta fase de gerenciamento da contratação é apresentada na SLTI como na Figura 4, e possui a atividade cíclica de “Encaminhar Ordem de Serviço” em paralelo com o subprocesso “Monitoramento da Execução”, no qual se realiza o controle interno da prestação de serviços.



**Figura 4 – Gestão do Contrato de TI (BRASIL, 2011a)**

Cruz et. al. (2011) definem ordem de serviço como um instrumento de controle das etapas de solicitação, acompanhamento, avaliação, atestação e pagamento de serviços, e que deve conter, no mínimo: definição e especificação dos serviços; volume e custo estimado dos serviços solicitados segundo as métricas definidas; resultados ou produtos solicitados e realizados; cronograma de realização dos serviços, incluídas todas as tarefas significativas e seus respectivos prazos; a avaliação de qualidade dos serviços realizados e as justificativas do avaliador; identificação dos responsáveis pela solicitação.

No sentido de orientar a gestão dos projetos nos órgãos públicos federais que o compõe, o SISP criou em 2011 uma Metodologia de Gerenciamento de Projetos, chamada MGP-SISP (BRASIL, 2011b), com bases fundamentadas no PMBoK 4ª Edição, na IN04 e na Lei nº 8.666, de 21 de junho de 1993.

Esta metodologia define 18 artefatos, dentre os quais estão o Documento de Oficialização da Demanda e Análise de Viabilidade do Projeto, ambos previstos na IN04, na fase de Planejamento da Contratação de TI, mas utilizados nesta metodologia individualmente para cada projeto de TI.

Em conformidade com o MGP-SISP (BRASIL, 2011b), lançou em 2013 a Metodologia de Gerenciamento de Portfólio de Projetos (MGPP-SISP) (BRASIL, 2013), no qual estabelece critérios de priorização dos projetos de TI.

## 2.3. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou parte do contexto legal das contratações públicas no Brasil, assim como as iniciativas para a melhoria desta no que concerne a contratação de soluções de tecnologia da informação.

Ante o exposto percebe-se grande rigidez quanto ao planejamento e fiscalização deste com uma estrutura robusta de fiscalização, tanto interna quanto externa e responsabilidade civil, inclusive penal para os envolvidos. Em contrapartida notam-se iniciativas de descentralização do poder (Decreto-Lei 200), embora estas ocorram apenas por meio formal específico com objetivo de aumentar a eficiência e a eficácia.

Percebe-se a importância da definição de critérios objetivos de qualidade usuais no mercado para avaliação dos serviços de software considerados comuns que resguardec legalmente o gestor do contrato.

Existem várias iniciativas de colaboração e orientação aos gestores de TI dos órgãos do SISP quanto ao processo de contratação de TI e o gerenciamento do mesmo. Sendo a *Ordem de Serviço* o documento central da execução contratual, que contém dentre outras coisas a definição dos itens que compõe o produto a ser recebido e seus critérios de aceitação.

---

## **CAPÍTULO 3 - *FRAMEWORK SCRUM***

### 3.1. CONSIDERAÇÕES INICIAIS DO CAPÍTULO

Este Capítulo apresenta uma síntese do *Framework Scrum*, sua criação, seu uso como metodologia de gestão de desenvolvimento ágil de software, e exemplos de aplicação. Apresenta ainda, algumas combinações do *Scrum* com outras metodologias de gestão como PMBoK e PRINCE2.

### 3.2. CRIAÇÃO DO SCRUM

Segundo Schwaber e Sutherland (2013), o *Scrum* é um *Framework* para desenvolver e manter produtos complexos que se baseia numa abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. Segundo Sutherland, Jakobsen e Johnson, (2008), o artigo intitulado “Novo jogo do desenvolvimento” de Takeuchi e Nonaka (1986) foi a base para a criação do *Scrum* em 1993 na Easel Corporation.

Takeuchi e Nonaka (1986) relataram as características comuns em times de alta produtividade no desenvolvimento de novos produtos em nove grandes empresas americanas e japonesas (Fuji-Xerox, Honda, Epson, 3M, Canon, NEC, Brother, Hewllet-Packard e Xerox). Como resultado, relacionaram as características identificadas com a formação *Scrum* no Rugby. Nessa formação todos os jogadores assumem suas posições de confronto e a bola é passada de um para outro numa jogada sincronizada e precisa em que tudo acontece de uma vez (*all-at-once*) e todos os integrantes estão ativos e cientes do movimento dos outros.

As seis características de equipes de alto desempenho relatadas são: (i) instabilidade embutida, (ii) times de projeto auto organizáveis, (iii) fases de desenvolvimento sobrepostas, (iv) multiaprendizado, (v) controle sutil e, (vi) transferência de aprendizagem organizacional (TAKEUCHI; NONAKA, 1986).

Beedle et al (1995) apresentam o *Scrum* de forma sistematizada como uma linguagem padrão para desenvolvimento hiperprodutivo de software. Esta foi a primeira vez que o *Scrum* foi apresentado formalmente embora já existisse e fosse utilizado em várias empresas, inclusive pelos autores.

Posteriormente, dois de seus autores (Ken Schwaber e Jeff Sutherland) iniciaram a publicação gratuita do *Guia Scrum*. Este guia possui dezoito páginas, e atualmente está traduzido para trinta e sete línguas (SCRUM.ORG, 2014). Sua última versão, publicada em

Julho de 2013, esclarece que o *Scrum* não é um processo ou uma técnica, mas um *Framework* dentro do qual vários processos ou técnicas podem ser empregados e que suas estratégias para uso variam e são descritas em outros documentos (SCHWABER; SUTHERLAND, 2013).

### 3.2.1. O PROCESSO DO *FRAMEWORK SCRUM*

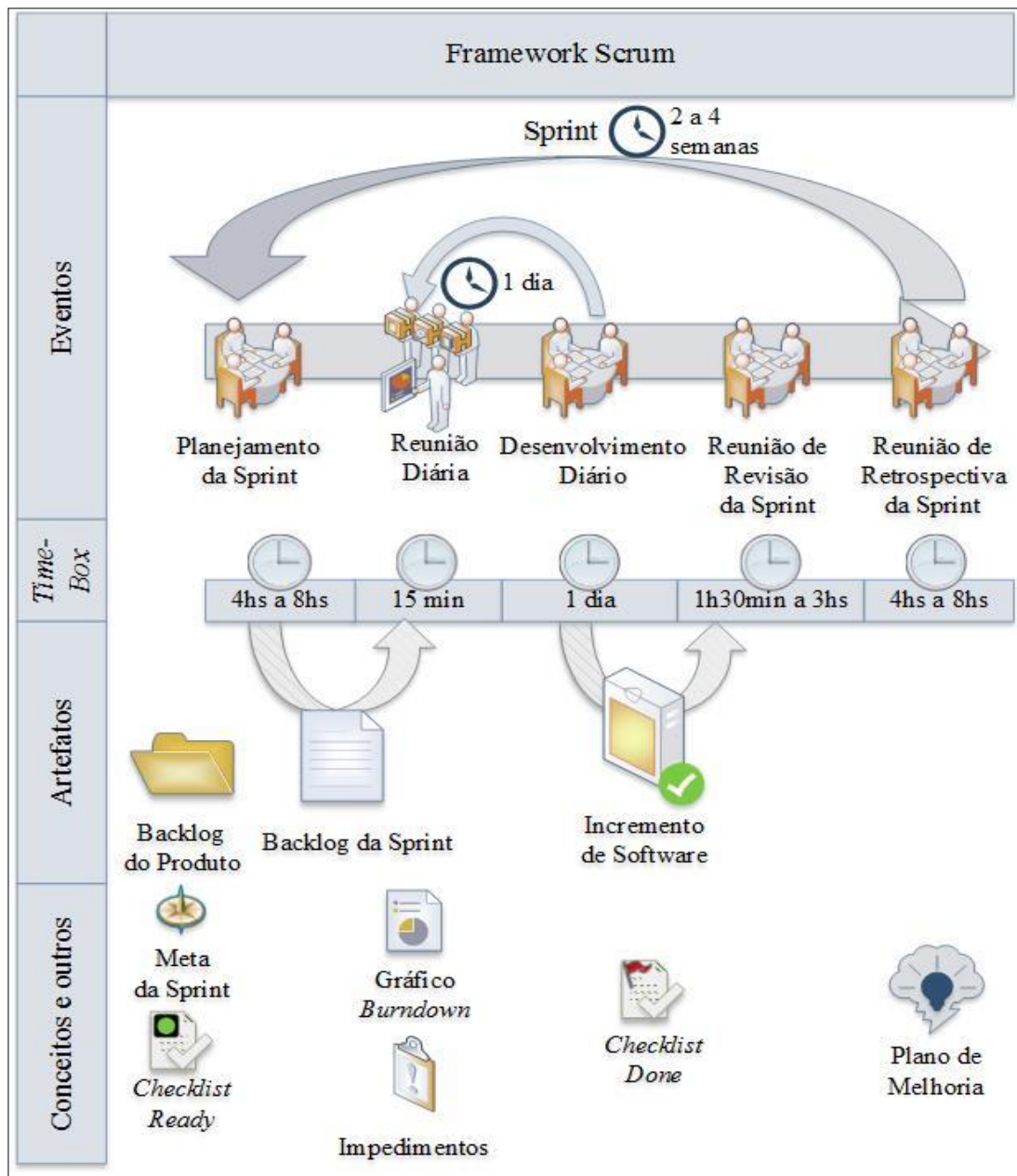
O *Framework Scrum* é composto por papéis, artefatos e eventos que se relacionam através de regras (SCHWABER; SUTHERLAND, 2013). Segundo Schwaber e Sutherland (2013), o *Scrum* se baseia em três pilares: Transparência, Inspeção e Adaptação, que significa que todos devem ter acesso a toda a informação e que os produtos devem ser inspecionados e adaptados conforme a necessidade.

Todos os eventos *Scrum* são *time-boxed*, de forma que todos possuem um tempo máximo de execução (SCHWABER; SUTHERLAND, 2013).

A essência do *Scrum* é a *Sprint*, nela são executados todos os outros eventos, seu artefato final é o Incremento de Software produzido pelo Time de Desenvolvimento a partir dos requisitos fornecidos no *Product Backlog* pelo *Product Owner*. Seu *time-box* é de duas a quatro semanas. O Guia *Scrum* define que uma *Sprint* pode falhar por não ter a sua meta satisfeita ao fim do seu *time-box* ou ser bem sucedida. Ainda a critério do PO ela pode ser cancelada para aproveitar oportunidades de negócio.

O *Scrum* se baseia no empirismo e nenhuma estimativa ou tendência deve ser mais forte que o histórico de *Sprints* anteriores (SCHWABER; SUTHERLAND, 2013). Por isso a primeira *Sprint* de um projeto, chamada de *Sprint Zero*, tem como principal objetivo permitir que o time de desenvolvimento se conheça e possa realizar melhores estimativas, assim como identificar os maiores riscos do projeto (SUTHERLAND; JAKOBSEN; JOHNSON, 2008a).

A Figura 5 apresenta os cinco eventos *Scrum* com seus *time-boxes* e seus três artefatos no fluxo de seu evento principal: a *Sprint*. Os outros símbolos dizem respeito a conceitos ou artefatos auxiliares citados no Guia *Scrum*, mas não considerados como artefatos, são eles os Conceitos de *Done* e *Ready*, a Meta da *Sprint*, os impedimentos, o Gráfico *Burndown*, e o plano de melhoria para a próxima *Sprint*.



**Figura 5 – Framework Scrum baseado no Guia Scrum (SCHWABER; SUTHERLAND, 2013)**

A Figura 6 apresenta seus três papéis fundamentais e únicos:



**Figura 6 – Composição do Time Scrum baseado no Guia Scrum (SCHWABER; SUTHERLAND, 2013)**

- *Product Owner (P.O.)* definido como o dono do produto é o único responsável por especificar todos os requisitos ao time, ninguém mais além dele tem autoridade para tomar decisões sobre o projeto. É responsável por manter, publicar e garantir o correto entendimento do artefato *Backlog* do Produto.
- *Time de Desenvolvimento* é composto por profissionais multifuncionais, como defendem Takeuchi e Nonaka (1986). São responsáveis por produzir o artefato *Incremento de Software* ao fim de cada *Sprint*. Se auto organizam e não possuem títulos oficiais entre si que não seja desenvolvedor.
- *Scrum Master* é o responsável por garantir o uso correto e constante das práticas definidas no *Framework Scrum*. Colabora com o *Time Scrum* para manter a agilidade propondo técnicas de gerenciamento do *Backlog* do produto, liderando o time para a alta produtividade e protegendo-os dos impedimentos que surgirem.

Seus artefatos definidos são três: O *Backlog* do Produto (BP), o *Backlog* da *Sprint* (BS) e o *Incremento de Software*, resultado final da *Sprint*. O *Backlog* do Produto é a única fonte de requisitos e se constitui de uma lista priorizada das funcionalidades do sistema mantida pelo *PO* em que cada item possui descrição ordem, estimativa e valor. O *Backlog* da *Sprint* é o recorte do BP para a *Sprint* juntamente com lista de tarefas e seus responsáveis. O escopo da *Sprint* não pode ser renegociado de forma que ponha em risco a meta da *Sprint*.

Pode-se considerar o *Backlog* do Produto, a fonte de requisitos e o *Sprint Backlog*, o planejamento para a realização daqueles requisitos em incremento de software. Todos estes devem ser transparentes para todo o *Time Scrum* que deve conhecê-los e ser capaz de inspecioná-los sempre que desejar. O Guia Scrum não apresenta exemplos ou *templates* para



a definição de nenhum artefato ou conceito definido (SCHWABER; SUTHERLAND, 2013).

A *Sprint* se inicia na Reunião de Planejamento que dura de quatro a oito horas dividida em duas partes na qual define-se o que pode ser realizado na *Sprint* e como isto será feito. Possui como entradas o *Backlog* do Produto, o último incremento de software, a capacidade projetada do Time de desenvolvimento e o objetivo pretendido pelo PO (SCHWABER; SUTHERLAND, 2013).

Acabada a etapa de negociação sobre o escopo e definida a Meta da *Sprint*, o Time de Desenvolvimento inicia o planejamento, definindo quais tarefas são necessárias, estimando o seu prazo e quem será responsável por elas no time. Ao definir o plano de entrega para os itens do *Backlog* do produto têm-se o *Backlog* da *Sprint* (SCHWABER; SUTHERLAND, 2013).

O Time de Desenvolvimento pode renegociar o escopo da *Sprint* com PO na Reunião de Planejamento da *Sprint*, sem que se altere a Meta da *Sprint* definida em conjunto. Como medida de verificação do planejamento o Time de Desenvolvimento precisa ser capaz de explicar aos outros integrantes (PO e SM) como transformará os itens do *Backlog* do produto pertencentes a *Sprint* em incremento que atenda ao objetivo (SCHWABER; SUTHERLAND, 2013).

As reuniões diárias ocorrem todos os dias, em 15 minutos, entre o Time de Desenvolvimento e é acompanhada pelo *Scrum Master*, com o objetivo de acompanhar o progresso rumo a Meta da *Sprint*. Cada um dos integrantes relata o que fez e o que fará no tempo de um dia que contribui ou contribuirá para o alcance da meta, assim como, declara os impedimentos encontrados. Segundo Sutherland e Schwaber (2013) o PO não participa dessas reuniões que é exclusiva do time de desenvolvimento (SCHWABER; SUTHERLAND, 2013).

Por fim, com o Incremento pronto, realiza-se a reunião de Revisão da *Sprint*. Esta não é uma reunião de status do projeto e sim uma reunião informal da qual participam o *Time Scrum* e os *Stakeholders* chave convidados pelo PO e possui *time-box* máximo de 4 horas (SCHWABER; SUTHERLAND, 2013).

Nesta revisão o PO apresenta o *Backlog* do Produto, identificando o que está pronto e o que não está e apresenta quando espera-se que o produto esteja finalizado. O Time de

Desenvolvimento por sua vez resume a *Sprint* apresentando os problemas surgidos e suas soluções, para então passar a apresentação do produto demonstrando o que está pronto de acordo com o conceito de *Done* definido. Dada a apresentação todos colaboram para a definição dos próximos passos, sendo possível realizar em conjunto análise de mercado, do contexto de uso do produto e suas mudanças e outras análises de tempo, orçamento, ganhos com o incremento para se definir o que esperar da próxima versão de acordo com o desejado e possível (SCHWABER; SUTHERLAND, 2013).

Ao fim desta reunião têm-se um *Backlog* do Produto revisado de forma a atender as necessidades e demandas dos clientes convidados (SCHWABER; SUTHERLAND, 2013).

A Retrospectiva da *Sprint* ocorre entre o fim da revisão e o início do planejamento da próxima *Sprint*. Possui um *time-boxed* de 3 horas para definir quais melhorias serão implantadas na próxima *Sprint* juntamente com seu plano de ação e revisar o conceito de *Done* para que este evolua de acordo com o aumento da capacidade do Time. Embora esta seja a oportunidade oficial para inspeção e adaptação do processo adotado pelo time, sua existência não impede que melhorias sejam executadas a qualquer momento se necessárias (SCHWABER; SUTHERLAND, 2013).

Relativo aos artefatos existem dois conceitos: *Ready* e *Done*, que significam preparado e pronto. O conceito de *Ready*, introduzido na última versão do Guia *Scrum*, diz respeito a maturidade dos requisitos presentes no BP para entrarem na *Sprint*. Já o conceito de *Done* diz respeito a conclusão destes itens ao fim da *Sprint*. O Guia *Scrum* não especifica quando são criados e apenas o conceito de *Done* é objetivamente mantido na Retrospectiva da *Sprint* (SCHWABER; SUTHERLAND, 2013).

Numa analogia com uma corrida, o *Ready* é o sinal verde para começar, e o *Done* é a bandeirada do final de cada volta, tanto o sinal verde como a bandeira estão apresentados na Figura 5.

O Guia do *Scrum* não estabelece atividades de engenharia de requisitos, monitoramento e controle ou de qualidade, embora relate que estes trabalhos devem ocorrer. A ausência de recomendações mais precisas condiz com as ideias de autogerenciamento e auto-organização do time *Scrum* e foco maior nas pessoas que nos processos como define o Manifesto para o desenvolvimento ágil de software (BECK et al., 2001a).

### 3.3. EMPREGOS DO SCRUM

Nesta Seção apresentam-se alguns empregos do *Scrum* relatados na bibliografia. Algumas iniciativas de adoção do *Framework* pelo poder público de outros países e do Brasil, com ressalvas as restrições do contexto burocrático e legal. Relata também os empregos do *Scrum* associados ao gerenciamento de produto e ao contexto de outsourcing.

#### 3.3.1. O SCRUM NO GOVERNO

As metodologias ágeis, têm sido introduzidas para uso no governo por suas promessas de entrega mais rápida com maior valor para o cliente e melhor qualidade (JAMIESON; VINSEN; CALLENDER, 2005a).

Jamieson; Vinsen; Callender (2005a) (2005b) propõem um método de contratação ágil utilizado no Departamento de Defesa Australiano para a manutenção de um sistema em constante evolução neste órgão. O método apresentado, consiste em dividir o ano em duas *releases* de seis meses cada, com *Sprints* de duas a quatro semanas, com usuários continuamente envolvidos de maneira formal e planejada e apresentações periódicas de protótipos. A estratégia de custeio é dividida em duas: tarefas simples com preço fixo estimado, e para tarefas de alto risco é realizada estimativa ao fim do design a partir da qual gera-se uma proposta de preço detalhada, sendo que em ambos os casos se o custo real em tempo, for menor que o estimado, o preço se ajusta para baixo.

Jamieson; Vinsen; Callender (2006) adicionam ao modelo de contratação o modelo de Valoração do Dinheiro baseado em *gates* de cada iteração do contrato, em que tanto o cliente, quanto o fornecedor pontuam um ao outro através de critérios estabelecidos e acordados para estabelecer de forma justa as expectativas e envolvimento das partes, destes dois valores compõem-se outro, todavia os autores resguardam o cálculo para outro artigo a ser publicado, que não foi encontrado.

Em 2011, o *Institute for Government*, uma entidade sem fins lucrativos com sede na Inglaterra, publicou um relatório sobre as falhas de TI no governo britânico (“*The Institute for Government*”, 2014). O relatório voltado a governança de TI relata que é necessária a adoção de metodologias ágeis no governo e propõe uma plataforma ágil que permita a colaboração entre os órgãos públicos de forma aumentar o retorno sob o investimento, dado que vários processos se repetem em diferentes órgãos.

Adicionalmente a esta proposta, em seus anexos, apresentam recomendações sobre a adoção de metodologias ágeis em contratações públicas, relatam onze estudos de caso com empresas privadas inglesas que adotaram ágeis, e os resultados de uma pesquisa com 18 representantes de órgãos governamentais ingleses sobre os desafios na implantação de ágeis.

Recomendam que para permitir flexibilidade no desenvolvimento e mudança dos requisitos requeridas pelas metodologias ágeis os contratos não devem ser feitos com base no escopo definido, que não existe no contexto ágil, mas na taxa de entrega da contratada, medida pelo *velocity*, com preço fixo sobre a combinação de tempo-materiais gastos. No caso da taxa de entrega não ser suficientemente rápida sugere o cancelamento do contrato.

O relatório alerta sobre a incompatibilidade entre as metodologias ágeis e o processo de revisões baseado em *gates* (INGLATERRA, 2007) definido pelo Escritório de Comércio do Governo inglês (*Office Government Commerce*). São relatados três principais motivos: o tempo estimado de execução do processo de revisão (8 semanas); a definição de *gates* baseado em um planejamento imutável e; a incapacidade do processo de aceitar falhas, como fazem os ágeis ao permitirem falhas pequenas e rápidas, com cancelamento de uma *Sprint* ou o completo descarte de seu resultado.

No ano seguinte, 2012, os órgãos do governo americano (GAO – *Government Accountability Office*) e britânico (NAO – *National Audit Office*) publicaram relatórios sobre o uso de metodologias ágeis, nos quais identificam melhores práticas e desafios para a adoção de ágeis no governo. Um ano depois, em 2013, o TCU publicou o Acórdão TCU 2314/2013 (BRASIL 2013) no qual avalia a utilização de metodologias ágeis no governo brasileiro. O Quadro 2 apresenta o objetivo, a identificação dos resultados e a metodologia utilizada em cada relatório/acórdão, adicionalmente identifica-se a presença do *Framework Scrum* nos mesmos.

Hajjdiab et. al (2012) apresentam oito motivos possíveis para a falha na adoção de metodologias ágeis em um estudo de caso de adoção do *Scrum* no governo dos Emirados Árabes com produção interna e partes do desenvolvimento terceirizado.

Aker et al (2013) relatam o estudo de caso da *Organização do Tratado do Atlântico Norte* (NATO/OTAN), de contratação por um processo sequencial e estrutura organizacional militar, fortemente estruturada, e executada com desenvolvimento ágil. Reportam nove lições aprendidas para o gerenciamento do projeto e sete para o desenvolvimento de software ágil.

	NAO (INGLATERRA, 2012)	GAO (EUA, 2012)	TCU (BRASIL, 2013)
Objetivo	Recomenda o uso de metodologias ágeis	Recomenda o uso de metodologias ágeis	Avalia o uso de metodologias ágeis
Resultados	Define 04 princípios para boa governança de TI com entregas ágeis e identifica 04 práticas comuns nos casos estudados.	Define 32 práticas e abordagens efetivas do uso de ágeis e 14 desafios	Caracteriza a contratação ágil em 05 órgãos públicos e identifica 16 riscos relativos a adoção de ágeis no governo federal
Metodologia de pesquisa	Avalia 07 estudos de caso de empresas privadas através de entrevistas semiestruturadas.	Realiza entrevistas estruturadas com 09 profissionais de mercado e valida resultados com 10 profissionais do governo.	Levantamento em seis órgãos públicos através de entrevistas e visitas técnicas com análise documental de cinco editais/contratos.
Uso do <i>Scrum</i>	Relata que a maioria dos órgãos públicos e privados utilizam o <i>Framework Scrum</i>	Não faz citação direta a metodologia mas utiliza do termo <i>Sprint</i> para designar as iterações	Apresenta o uso do <i>Framework Scrum</i> em todos os casos estudados.

**Quadro 1 – Caracterização dos relatórios sobre metodologias ágeis no governo**  
(Autor)

A partir da análise das práticas recomendadas encontradas nos casos relatados este autor extraiu cinco categorias: adoção de ágeis, capacitação das pessoas, planejamento, requisitos e monitoramento e acompanhamento apresentadas no Quadro 3.

Adoção de ágeis	<ol style="list-style-type: none"> <li>1. Definir uma estratégia de adoção ágil, melhorando continuamente a adoção de “agilidade” tanto no nível do projeto quanto no nível da organização. (EUA, 2012) adaptando se necessário porém sem comprometer a eficiência e eficácia (HAJJDIAB; TALEB; ALI, 2012);</li> <li>2. Planejar um projeto piloto de tamanho médio, não crítico. Ter o apoio de um mestre ágil (<i>coach</i> experiente em desenvolvimento ágil) e diminuir aceitação de novos projetos por seis meses, para trabalhar na mudança de comportamento (HAJJDIAB; TALEB; ALI, 2012);</li> <li>3. Ter gerentes de nível médio para orientar o time de desenvolvimento assim como gerentes de garantia da qualidade experientes com autoridade e participante do time (AKER et al., 2013) e;</li> <li>4. Ter áreas de trabalho colaborativas (INGLATERRA, 2012)</li> </ol>
Capacitação	<ol style="list-style-type: none"> <li>1. Melhorar a migração do processo então vigente para os conceitos ágeis usando termos e exemplos ágeis. (EUA, 2012);</li> <li>2. Capacitar equipes pequenas, multifuncionais. (EUA, 2012);</li> <li>3. Deve-se prever a evolução do time e não esperar a perfeição na primeira iteração (HAJJDIAB; TALEB; ALI, 2012);</li> <li>4. Ter o mestre ágil (<i>coach</i> experiente em desenvolvimento ágil), contratar alguns profissionais experientes em ágeis, ministrar constantemente cursos ágeis (HAJJDIAB; TALEB; ALI, 2012).</li> </ol>
Planejamento	<ol style="list-style-type: none"> <li>1. Planejar a participação do usuário e prever a necessidade de participação de especialistas no negócio (AKER et al., 2013);</li> <li>2. Definir uma estratégia para gerenciar os requisitos e design que devem estar totalmente definidos antes da <i>Sprint</i> (AKER et al., 2013);</li> <li>3. Planejamento da iteração e da <i>Release</i> (INGLATERRA, 2012);</li> </ol>

Requisitos	<ol style="list-style-type: none"> <li>1. Obter feedback dos interessados/cliente com frequência e de perto. (EUA, 2012);</li> <li>2. Incluir requisitos relacionados com a segurança e monitoramento do progresso na sua fila de trabalhos não concluídos (<i>Backlog</i>) (EUA, 2012);</li> <li>3. Diminuir a quantidade de documentação não necessária, com a aprovação da alta gerência (HAJJDIAB; TALEB; ALI, 2012);</li> <li>4. Conflito entre requisitos ágeis e especificação tradicional deve ser gerenciado pelos gerentes de projeto (AKER et al., 2013);</li> <li>5. Ao desenvolver com metodologia ágil em um ambiente de contratação "cascata" utilizar as fases planejadas para estudar os requisitos em profundidade, capacidade por capacidade, a cada iteração (AKER et al., 2013);</li> <li>6. Com o detalhamento dos requisitos por fase, o nível de detalhe na transferência de conhecimento deve ser gerenciado (AKER et al., 2013)</li> <li>7. A priorização do escopo do projeto deve ser desenvolvida inicialmente e acompanhada com rigorosos padrões de qualidade para aceitação (AKER et al., 2013);</li> <li>8. Realizar desenvolvimento centrado na interface do usuário garante que usuários operacionais estejam envolvidos na implementação do sistema (AKER et al., 2013);</li> <li>9. Ambos, contratada e contratante, devem garantir que os resultados dos <i>workshops</i> de apresentação de interface do usuário são consistentes com os requisitos contratuais e padrão visual da empresa (AKER et al., 2013);</li> </ol>
Monitoramento e Acompanhamento	<ol style="list-style-type: none"> <li>1. Procurar identificar e resolver os impedimentos nos níveis da organização e do projeto (EUA, 2012);</li> <li>2. Ganhar a confiança, demonstrando valor, no final de cada iteração (EUA, 2012);</li> <li>3. Reuniões Diárias (<i>daily stand-up meetings</i>) (EUA, 2012) (INGLATERRA, 2012);</li> <li>4. Acompanhar o progresso utilizando ferramentas e métricas (EUA, 2012) (INGLATERRA, 2012);</li> <li>5. Manter métricas e quadro de atividades visíveis. (EUA, 2012) (INGLATERRA, 2012);</li> <li>6. Ao equilibrar as filosofias ágeis e cascata os Gerentes de Projeto devem especificar quais detalhes serão definidos na fase de desenvolvimento ágil e o que precisa ser acordado na revisão crítica do design (AKER et al., 2013);</li> <li>7. Novos produtos são necessários no CDR (<i>Critical Design Review</i>) para gerenciar o desenvolvimento ágil em um contrato cascata, por exemplo, um <i>Roadmap</i> da <i>Sprint</i> (AKER et al., 2013);</li> <li>8. É necessário rastrear com acurácia a porcentagem de completude do design e do desenvolvimento com o objetivo de orientar a decisão de <i>Trade-off</i> entre o escopo e qualidade quando o cronograma é fixo (AKER et al., 2013);</li> </ol>

**Quadro 2 - Práticas Recomendadas em Casos Governamentais (Autor)**

Para a seleção de fábricas de software/times ágeis por órgãos públicos europeus Pries-Heje; Pries-Heje; Dalgaard (2013) propuseram o *Code Camp Scrum*, baseada no *Scrum* e no evento *Code Camp*<sup>3</sup>. A proposta é uma espécie de competição técnica, com preferencialmente um dia de duração, entre as fábricas/times-ágeis, composta de seis partes: Script; *Product Backlog*, Papéis, Infraestrutura Técnica e Avaliação, a qual foi experimentada em dois estudos de caso com sete times diferentes apresentando bons resultados. O que muda o processo de seleção burocrático definido anteriormente.

Moe & Newman (2014) apresentam a análise de três estudos de caso de contratos de fábrica de software em órgãos públicos noruegueses, entre 2012 e 2013. Eles estabelecem como desafios do contexto da contratação pública a necessidade de definição prévia dos requisitos; o gerenciamento adequado dos *Stakeholders*, que deve considerar a inserção tardia de novos *Stakeholders* com visões de negócio distintas; a presença de usuários apegados ao sistema legado sendo substituído e; a falta de definição de regras de decisão o que levou nos casos estudados a discussões inconclusivas.

Em todos os casos a rigidez contratual relacionada ao escopo é relatada como limitação do contexto, assim como a dificuldade em garantir tanto o comprometimento da área de negócio com o projeto quanto a compreensão e suporte de todos os envolvidos a filosofia ágil, mais do que apenas as práticas ágeis.

### 3.3.2. ESTRATÉGIA DE EMPREGO DO SCRUM

*“Estratégias específicas para o uso do Framework Scrum variam e são descritas em outros documentos.” (SCHWABER; SUTHERLAND, 2013)*

Esta seção apresenta algumas estratégias encontradas na literatura. Sendo o *Scrum* um *Framework* extremamente difícil de dominar, Sutherland (2005) revisita o esquema de sobreposição das fases de desenvolvimentos apresentado por Takeuchi e Nonaka (1986) como três tipos de implementação do *Scrum*, apresentado na Figura 7 – Estratégias de Execução das *Sprints* adaptado de (SUTHERLAND, 2005) (TAKEUCHI; NONAKA, 1986).

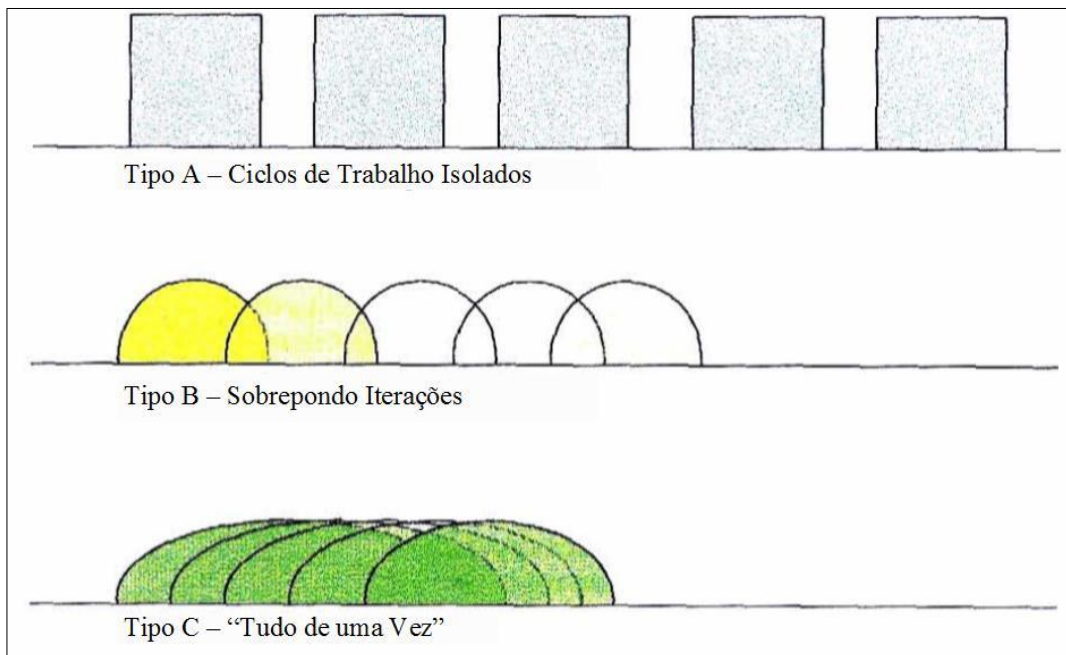
O Tipo A é o mais simples comum para iniciantes no qual existe um intervalo de tempo entre as *Sprints* que idealmente seria o mesmo, porém na prática varia bastante como

---

<sup>3</sup> Code Camp é um evento comunitário no qual desenvolvedores aprendem com outros desenvolvedores (“SiliconValley CodeCamp”, 2014)



relatado por Sutherland; Jakobsen e Johnson (2008b). Os autores citam uma tentativa falha de aplicação do *Scrum* em que os intervalos para integração dos produtos ficavam cada vez maiores, apresentando deterioração do ritmo de desenvolvimento.



**Figura 7 – Estratégias de Execução das *Sprints* adaptado de (SUTHERLAND, 2005) (TAKEUCHI; NONAKA, 1986)**

O tipo B é pouco sobreposto excluindo o intervalo entre uma *Sprint* e outra. Neste estágio o *Product Backlog* da *Sprint* n+1 é preparado ao longo da *Sprint* n, o que leva a necessidade de documentação de requisitos de forma minimalista, com o suficiente para a próxima *Sprint*, podendo ter pequenas sobreposições pontuais em outras áreas.

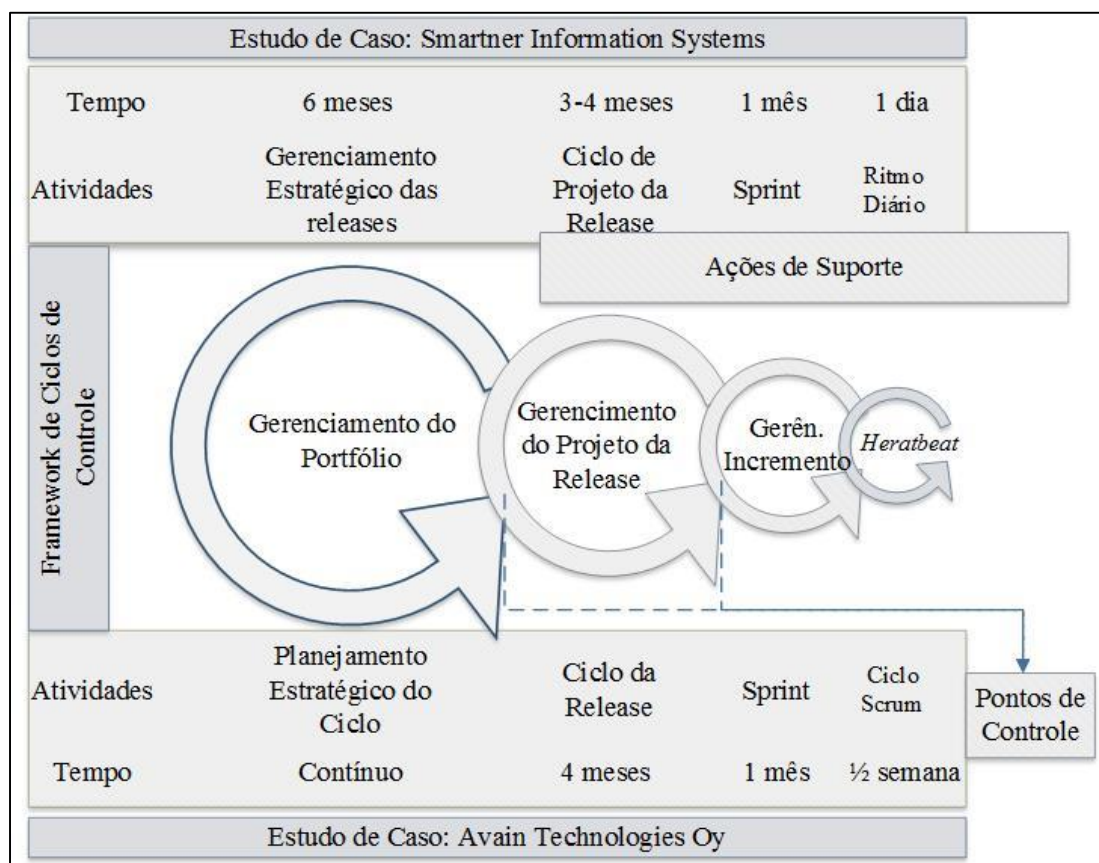
O tipo C é completamente sobreposto e no qual se encontra a maior produtividade, nele a sobreposição das atividades é total e a forma como o time se organiza é proativa e completamente instável podendo ser alterada ao longo de todo o projeto, neste tipo o planejamento é constantemente revisto (SUTHERLAND, 2005) (TAKEUCHI; NONAKA, 1986). Sutherland (2005) relata neste estágio a resolução de problemas técnicos previstos para a *Sprint* seguinte permitindo, assim maior velocidade.

Para o controle organizacional e gerenciamento de produtos de software de empresas de pequeno porte e baseado no *Framework Scrum* e no *Extreme Programming* (XP) Rautiainen; Lassenius; Sulonen (2002) criaram o *Framework* de Ciclos de Controle (FCC)

na Universidade de Tecnologia de Helsinki (Finlândia). Este *Framework* define quatro ciclos de gerenciamento de forma abstrata: Portfólio, *Release*, Incremento e *Heartbeat*.

Foram relatados estudos de caso bem sucedidos em duas fábricas de software finlandesas: *Avain Technologies Oy* (VANHANEN; ITKONEN; SULONEN, 2003) e *Smartner Information Systems* (RAUTIAINEN; VUORNOS; LASSENIUS, 2003a), ambas estratégias de uso apresentadas na Figura 8 – *Framework* de Ciclos de Controle com *Scrum* adaptado de (VANHANEN; ITKONEN; SULONEN, 2003) (RAUTIAINEN; VUORNOS; LASSENIUS, 2003a).

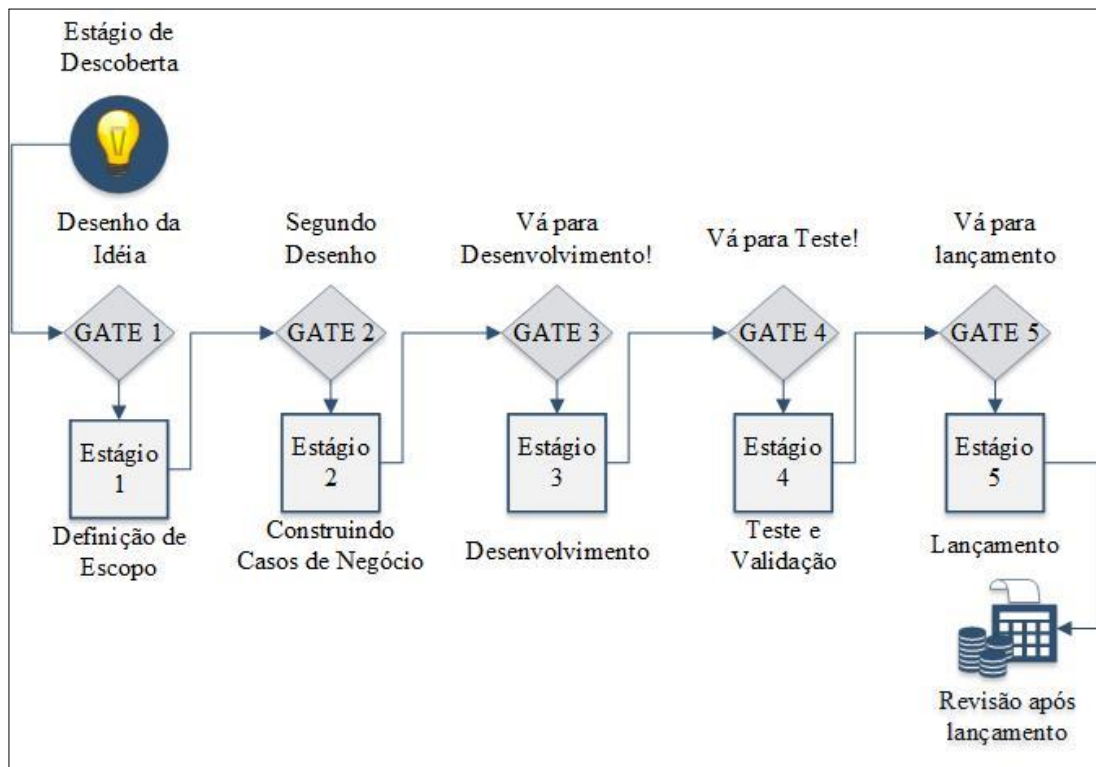
Nota-se a diferença entre os *time-boxes* de cada ciclo de acordo com o contexto. O *heartbeat*, que é o menor ciclo de acompanhamento do projeto, muda de 1 dia a ½ semana, dependendo da disponibilidade do cliente em cada caso, assim como a estratégia de acompanhamento, que em um caso é realizada através de ações de suporte e no outro em pontos de controle.



**Figura 8 – Framework de Ciclos de Controle com Scrum adaptado de (VANHANEN; ITKONEN; SULONEN, 2003) (RAUTIAINEN; VUORNOS; LASSENIUS, 2003a)**

Para o controle e monitoramento do desenvolvimento de produtos Cooper (1990) propôs o modelo *Stage-Gate* apresentado na Figura 9 – Modelo *Stage-Gate* adaptado (COOPER, 1990) se baseia em portas de decisão (*Gates*) na qual a diretoria da organização pode aprovar a continuação, ordenar a repetição da fase anterior ou cancelar o projeto.

Ao relacionar este modelo com o *Scrum*, Cohn (2010) esclareceu que ele se aplica a governança do projeto, pois visa alinhar o projeto aos objetivos da organização. Estabelece ainda a diferença entre a governança do projeto e o gerenciamento do projeto realizado nos eventos *Scrum* sem a participação da diretoria, e defende que o Time *Scrum* deve para organizar o seu gerenciamento sem interferência da Diretoria.



**Figura 9 – Modelo *Stage-Gate* adaptado (COOPER, 1990)**

Em concordância Sutherland (2005) defende que inserir gerenciamento externo no time *Scrum*, pode causar perda de autonomia do time, altas taxas de defeitos, excessivas horas extras, esgotamento do pessoal e alta rotatividade de funcionários, dada a pressão para aumento de produtividade da supervisão externa.

Outro modelo para gestão de produtos é o PRINCE2 (**P**ROjects **I**N **C**ontrolled **E**nvironments ou Projetos em ambientes controlados) que segundo Sánchez; Gaya & Pérez

(2013) é uma versão britânica do PMBoK criada pelo Escritório de Comércio do Governo (*Office of Government Commerce*). Porém seu site o define como metodologia de gerenciamento de produto diferenciando-o do PMBoK que é um corpo de conhecimento geral que não possui processo inserido (COURSE, 2013).

O PRINCE2, é estruturado em sete princípios, temas, seis métricas de monitoramento do projeto e seu processo em estágios de controle (TURLEY, 2010) similar ao definido por Cooper (1990). Seus princípios são orientados aos produtos o que lhe confere concordância com o manifesto ágil (BECK et al., 2001).

Esta metodologia estabelece ainda três níveis de gerenciamento, sendo o time o nível mais inferior, responsável por desenvolver os produtos e pelo gerenciamento diário; o mais superior o Comitê do Projeto, responsável pelas decisões de continuidade do projeto, como sugere Cooper (1990) e o do projeto como nível intermediário responsável por coordenar as ações entre os dois níveis e por retirar os impedimentos relatados pelo Time (TURLEY, 2010).

Schwalbe (2012) relacionou as atividades comuns do *Scrum* nas fases de gerenciamento estabelecidas no PMBoK como apresentado no Quadro 3 – *Scrum* e PMBoK adaptado de (SCHWALBE, 2012), sendo que no encerramento encontra-se apenas a reunião de retrospectiva da *Sprint*.

Fase	Atividades
Iniciação	<ul style="list-style-type: none"> <li>• Determinar os papéis;</li> <li>• Decidir quantas <i>Sprints</i> vão compor cada <i>release</i> e o escopo do software a ser entregue;</li> </ul>
Planejamento	<ul style="list-style-type: none"> <li>• Criar o <i>Product Backlog</i>;</li> <li>• Criar o <i>Sprint Backlog</i>;</li> <li>• Criar <i>Backlog</i> da <i>Release</i>;</li> <li>• Planejar cada o trabalho a cada dia na Reunião Diária;</li> <li>• Documentar impedimentos em uma lista;</li> </ul>
Execução	<ul style="list-style-type: none"> <li>• Conclua tarefas a cada dia durante a <i>Sprint</i>;</li> <li>• Produzir um Incremento de Software a cada <i>Sprint</i>;</li> </ul>

Monitoramento e Controle	<ul style="list-style-type: none"><li>• Resolver problemas e impedimentos;</li><li>• Criar e atualizar gráfico <i>Burndown</i>;</li><li>• Demonstrar o produto completo durante a reunião de revisão da <i>Sprint</i>;</li></ul>
Encerramento	<ul style="list-style-type: none"><li>• Refletir sobre a forma de melhorar o produto e o processo, durante a Reunião de Revisão da <i>Sprint</i>;</li></ul>

**Quadro 3 – Scrum e PMBoK adaptado de (SCHWALBE, 2012)**

Sutherland; Harrison e Riddle (2014) apresentam padrões associados ao uso do *Scrum* criados e validados pela comunidade *Scrum* PLoP<sup>4</sup> e relataram os seis padrões mais eficientes, apresentados no Quadro 4 – Os melhores padrões *Scrum* adaptado de (SUTHERLAND; HARRISON; RIDDLE, 2014).

---

<sup>4</sup> *Scrum* PLoP é uma comunidade de usuários do *Scrum*, criada por Jeff Sutherland, para compartilhar experiências de práticas complementares ao *Scrum* e permitir sua evolução. O conceito de Ready começou com um padrão *Scrum* PLoP e depois foi incorporado no Guia *Scrum*. (“Published Patterns”, 2014). < <http://www.scrumplop.org/>>

Padrão <i>Scrum</i> PLoP	Descrição
Times estáveis	Manter as mesmas pessoas nos times as capacita para realizar estimativas de desenvolvimento mais acuradas.
Águas Passadas	“Seja o Empírico!” Na maioria dos casos a estimativa de produção da última <i>Sprint</i> é o melhor indicador.
Todos em um pedaço	Foque o máximo de esforço em uma História por vez. Eleja um capitão para esta história, que não pode ser interrompido e deve seja auxiliado por todos sempre que necessário.
Padrão de Interrupções: Interrupções não legítimas	Determina um <i>buffer</i> de interrupções (horas ou dias) para novos itens adicionados pelo PO ou para requisitos mal planejados, ou para correções de <i>Sprints</i> passadas. Se o <i>buffer</i> estourar deve ser realizado replanejamento.
Código Limpo diariamente	Mantenha o código limpo sem bugs e permita uma evolução consistente. Qualidade é tão importante quanto quantidade.
Procedimento de Emergência	Quando o Time percebe que as coisas não estão indo como previstas, geralmente pelo gráfico Burndown, realizam o procedimento de emergência até o meio da <i>Sprint</i> . O <i>Scrum Master</i> deve intervir se não acontecer.
<i>Scrumming</i> o <i>Scrum</i>	Transforme a melhoria em História com testes de aceitação e insira-a no <i>Backlog</i> da <i>Sprint</i> seguinte. A retrospectiva deve ser rápida e voltada a solução e não aos problemas.
Métrica da Felicidade	É um indicador preditivo a ser utilizado com o Time de Desenvolvimento, se estiverem tristes significa que algo deve estar errado e o <i>Scrum Master</i> deve intervir.
Times que terminam antes aceleram mais rapidamente	Não seja otimista, coloque pouco trabalho dentro da <i>Sprint</i> e termine o mais rápido possível.

**Quadro 4 – Os melhores padrões *Scrum* adaptado de (SUTHERLAND; HARRISON; RIDDLE, 2014)**

A maioria deve ser utilizada dentro do desenvolvimento pelo Time de Desenvolvimento, mas em contexto de *outsourcing* podem ser utilizados os padrões “Times que terminam antes aceleram mais rapidamente”, “Águas Passadas” e “Padrão de

Interrupções: Interrupções não legítimas” que orientam a conduta do Time *Scrum* no Planejamento da *Sprint*; e o “*Scrumming* o *Scrum*” que determina apenas uma melhoria a ser tratada por vez em cada *Sprint*, reduzindo o tempo da Retrospectiva da *Sprint* e possibilitando realizá-la como parte da Reunião de Revisão da *Sprint*

O padrão “Procedimento de Emergência” depende do Time de Desenvolvimento e *Scrum Master* mas reforça o monitoramento organizacional apenas no nível de produto. Embora a Métrica da Felicidade seja um indicador efetivo a respeito do sucesso da *Sprint*, seu uso em um ambiente em que o contrato é no mínimo tão importante quanto a colaboração fica comprometido pela característica de gerenciamento externo condenada por Sutherland (2005).

Alinhado a definição e controle do produto em níveis apresentados pelo FCC (RAUTIAINEN; LASSENIUS; SULONEN, 2002), e pelo PRINCE2 (TURLEY, 2010) Davis (2013) propõe a definição de *Done* incremental nos níveis de História, *Sprint* e *Release*. Sendo que o *Done* da História ocorre ao longo da *Sprint*, a cada dia ou dois, até a conclusão de cada História, e o da *Sprint* até o fim desta, sendo o *Done* da *Release* resultado da última *Sprint* da *Release* que deve contemplar o incremento da definição de *Done*. Este estudo de caso também relata a melhoria e o aumento da definição de *Done* ao longo das *Sprints* além da evolução estruturada nos níveis de controle.

Em um estudo de caso na *Patient Keeper*, uma empresa hospitalar americana, Jakobsen & Sutherland (2009), relatam o uso *releases* mensais de duas *Sprints*, pois a empresa não conseguiu absorver os produtos de forma mais rápida.

Porém, mesmo adotando esta estratégia ressaltam a importância da disponibilização do Incremento de Software a cada *Sprint* de duas semanas para validação. O mesmo argumento é defendido por Gallardo-Valencia & Sim (2009) que mapeiam a validação em metodologias ágeis ocorrendo através das práticas de colaboração e entrega contínua do Incremento de Software.

Para a definição dos requisitos VLAANDEREN et al (2011) propõem um Processo Ágil de Gerenciamento de Produtos de Software baseado no *Scrum* que prevê a preparação do *Backlog* do Produto até a metade da *Sprint* de desenvolvimento. Essa estratégia situa-se no Tipo B de execução da *Sprint* proposto por Sutherland (2005) e apresentada na Figura 7 – Estratégias de Execução das *Sprints* adaptado de (SUTHERLAND, 2005) (TAKEUCHI; NONAKA, 1986).

Porém a proposta não inclui a participação do Time de Desenvolvimento na *Sprint* de Gerenciamento de Produto. A diferença entre a *Sprint* proposta e a de Desenvolvimento é apresentada no Quadro 5 – Diferenças entre *Scrum* de Desenvolvimento e de Gerenciamento de Produto Ágil adaptado de (VLAANDEREN et al., 2011). Em ambos os casos o mesmo *Backlog* do Produto é a origem do trabalho, mas mudam as entradas, saídas e seus responsáveis.

	<i>Backlog</i> da <i>Sprint</i> do Gerenciamento do Produto	<i>Backlog</i> da <i>Sprint</i> de Desenvolvimento
Origem do trabalho	<i>Backlog</i> do Produto	<i>Backlog</i> do Produto
Entrada	Visão (requisitos não especificados) Bugs	Requisitos Especificados
Saída	Requisitos Especificados	Software Funcional
Objetos Tratados	Visões, conceitos, Temas, requisitos	Bugs, melhorias de produto, funcionalidades, tecnologia, evoluções, etc.
Organização do trabalho	<i>Sprints</i> e Reuniões diárias	<i>Sprints</i> e Reuniões Diárias
Responsável	Gerentes de Produtos	Desenvolvedores
Resultado no BP	Definições de Requisitos	Itens do BP Finalizados

**Quadro 5 – Diferenças entre *Scrum* de Desenvolvimento e de Gerenciamento de Produto Ágil adaptado de (VLAANDEREN et al., 2011)**

Leffingwell (2007) (2011) apresentou o Scaled Agile Framework (SAFe) como uma evolução dos processos (ágeis) adaptativos a um nível empresarial, no qual define três níveis de planejamento (portfólio, programa, time) e para estes níveis definiu papéis, artefatos e eventos, utilizando no nível de Time conceitos próprios do Scrum como Product Owner e Sprint.

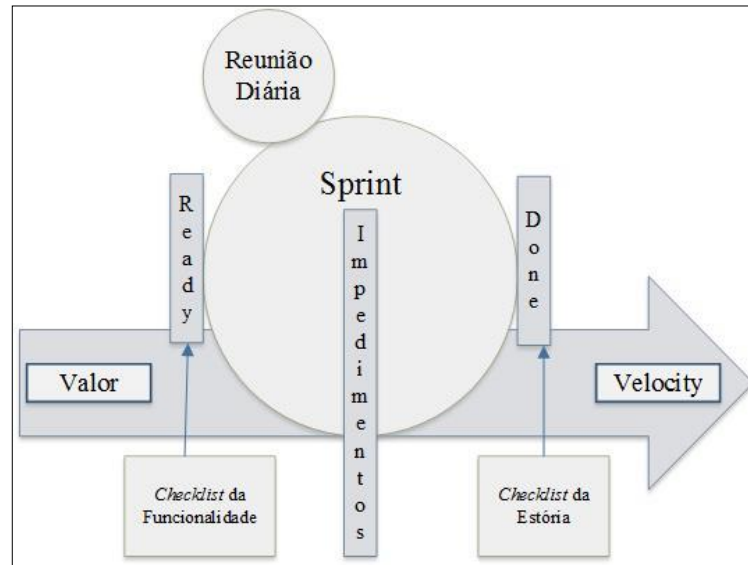


Como possui estreita relação com o Scrum Leffingwell (2013b) apresenta duas adaptações: uma Sprint de validação e a HIP (*Hardening – Innovation – Planning*) Sprint, com Backlog vazio, para realizar atividades que não poderiam ser feitas em cada Sprint como a criação de documentos de usuário, validação da release com seus critérios específicos e criação do pacote de entrega.

A HIP *Sprint* ainda permite a realização de atividades que deveriam ser feitas antes e que o Time de Desenvolvimento não conseguiu fazer como a execução dos testes de integração, testes dos requisitos de sistema, integração final do sistema, eliminação de defeitos, entre outros porém quanto maior for a *Sprint* HIP, menos “ágil” e mais sequencial se torna o desenvolvimento (LEFFINGWELL, 2013c).

Esta estratégia de execução dos releases pode ser comparada ao Tipo A de estratégias de execução das *Sprints* (apresentada na Figura 8) com intervalo para atividades de ajustes e conclusão do desenvolvimento entre uma iteração e outra.

Jakobsen e Sutherland (2009) esquematizam o fluxo de trabalho do *Scrum* como apresentado na Figura 10 – Fluxo de Trabalho do *Scrum* adaptado de (JAKOBSEN; SUTHERLAND, 2009), como entrada o valor para o negócio que deve passar porta de qualidade do *checklist* de *Ready* para iniciar o desenvolvimento, no qual se responde aos impedimentos e ao fim o produto de software passa pela porta de qualidade do *checklist* de *Done* resultado na taxa de entrega medida pelo *Velocity* que é a taxa de entrega do software medida pelo time.



**Figura 10 – Fluxo de Trabalho do Scrum adaptado de (JAKOBSEN; SUTHERLAND, 2009)**

Dado o planejamento das *Releases*, estudos de caso (AKER et al., 2013) (SUTHERLAND et al., 2008) (JAKOBSEN; SUTHERLAND, 2009) apresentam a definição do conceito de *Ready*, o uso da definição de protótipos de interface gráfica e avaliação da arquitetura e impedimentos tecnológicos, relatando todos bons resultados.

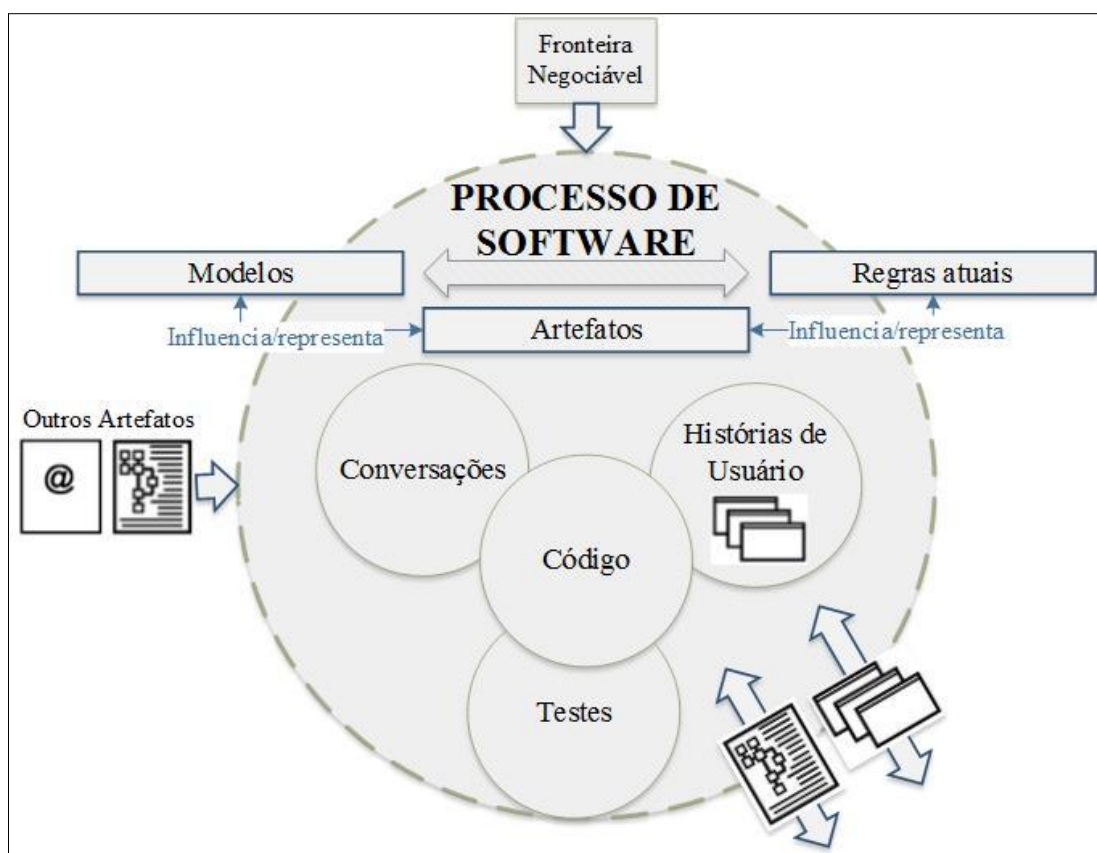
Cohn, Sim e Lee (2009) avaliaram o relacionamento dos artefatos no processo de software em duas empresas que adotaram o *Scrum*. Utilizaram entrevistas, observação e coleta de documentos. Como resultado definem a fronteira do processo de software como sendo negociável ao longo do desenvolvimento através do que chamam de “A Conversação”, que são as conversas usuais do dia-a-dia do desenvolvimento de software e dos eventos *Scrum*.

A Figura 11 – Processo de Software apresenta o modelo conceitual proposto por Cohn, Sim e Lee (2009) (adaptado) que estabelece artefatos que podem estar tanto dentro, como fora, e cruzando a fronteira negociável do Processo de Software. Definem as conversações como artefato e, estabelecem os modelos e as regras atuais (*enactments*) representados e influenciados por estes artefatos, como constituintes do processo de software, que é vivo, e se ajusta as necessidades do software sendo produzido, podendo mudar muito entre uma *Sprint* e outra.

A pesquisa define como principais artefatos internos o código com os testes as Histórias de Usuário que compõe do *Backlog* da *Sprint* e as conversações; os que cruzam a

fronteira são as Histórias de Usuário do *Backlog* do Produto e outros artefatos como padrões de arquitetura, referências de sistemas externos envolvidos e, outros; dentre os externos estão os contratos e a negociação de escopo do que entrará para o Processo de Software.

Na pesquisa Cohn, Sim e Lee (2009) encontraram artefatos criados pela especialista de teste do time e pelo *Compliance Officer* no Time, que era o responsável pela conformidade contratual nos projetos. Identificaram estes artefatos como resquícios de suas experiências com uma metodologia orientada a documentação.



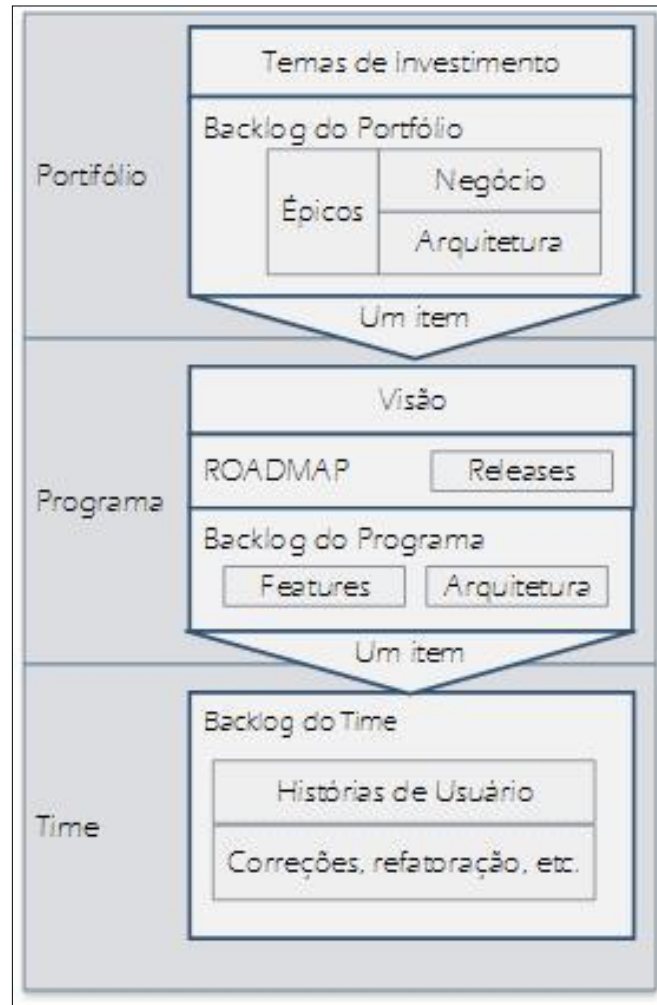
**Figura 11 – Processo de Software Ágil adaptado de (COHN; SIM; LEE, 2009)**

O *Compliance Officer* criou termos de aceite de histórias de usuário para assinatura que nunca foram assinados e planilhas de rastreabilidade de mudança nas histórias de usuário, para acompanhar “quem” mudou “o quê”, “quando” e “por quê” que também não eram usadas a não ser por ele nos eventos *Scrum*. Já a especialista de testes criou uma Wiki com a definição dos testes de cada *Sprint*, incluindo testes de aceitação.

Ambos julgaram estes artefatos como desperdício de tempo pois mantinham um histórico de informações que consideravam desnecessário no *Scrum* e não geravam valor direto para o software. Todavia a Wiki de teste era usada esporadicamente por outros desenvolvedores quando estes pediam explicações sobre os testes realizados em *Sprints* passadas, mas a maioria nem sequer sabia que o artefato existia.

Coallier (2007), Heeager e Nielsen (2009) defendem que é possível conciliar uma abordagem orientada a documentação oriunda da necessidade de ajuste a padrões, normas e leis, mesmo que hajam dificuldades introduzidas pelo gerenciamento dos documentos. A existência destes requer apenas maior disciplina e competência dos engenheiros e não inviabilizam o uso de outras práticas ágeis, que podem manter o ritmo de desenvolvimento.

Leffingwell (2013a) define no *Scaled Agile Framework* três tipos de *Backlog* de acordo com o nível organizacional e de detalhamento como apresentado na Figura 12 – Estrutura do *Scaled Agile Framework* adaptado de (LEFFINGWELL, 2013b). Define ainda que o *Backlog* do Time (similar ao *Backlog* da *Sprint*) possui dois tipos de itens Histórias de Usuário e outro que reúne Correções, refatorações e outras intervenções no código.



**Figura 12 – Estrutura do *Scaled Agile Framework* adaptado de (LEFFINGWELL, 2013b)**

Cohn (2010) ao contrário apresenta a organização do *Backlog* do Produto apenas com histórias de usuário de tamanhos distintos indo de Épicos a histórias com escopo menos e mais claras a nível do *Backlog* da *Sprint*. O mesmo diz Schwaber (2004) que apresenta em seu esquema modelo de *Backlog* do Produto os itens prioritários sendo as histórias menores e mais claramente definidas sendo seguidas pelas histórias da *Sprint* seguinte e das que representam as *releases*.

Na mesma linha Fowler (2006) argumenta, em concordância com BATRA (2009) que quanto menor for a disponibilidade de interação entre o Time *Scrum* (*Product Owner* , Equipe de Desenvolvimento e *Scrum Master*) maior será a necessidade de documentação que deve ser criada, porém ressalta que essa documentação deve ser orientada ao código para ser inserida dentro do processo de software. Para seguir esta estratégia sugere o uso de testes de aceitação como parte da documentação adicional necessária.

De modo a permitir a possibilidade de inovação Rautiainen; Vuornos & Lassenius (2003b) propuseram em seu estudo de caso a piscina de ideias (*pool of ideas*) que é uma lista de ideias e inovações sobre o produto que poderia ser utilizada por todos os *Stakeholders* e seria avaliada pelo PO juntamente com o grupo responsável pelo orçamento do projeto. Embora esta ferramenta só tenha sido utilizada uma vez no estudo de caso apresentado, consideraram uma boa prática e justificam seu baixo uso graças a visão compartilhada das soluções necessárias e ao ambiente carente no qual não havia espaço para inovações.

Yi (2011) defende o uso do gerente de projetos como *Scrum Master* porém Cohn (2010) realça os perigos dessa abordagem relatando quatro tipos de gerentes diferentes sendo o *Scrum Master* o tipo oposto do Gerente de Projetos funcional que no extremo controlará apenas as tarefas (através do Gráfico *Burndown*) enquanto o *Scrum Master* é um facilitador que protege o time de interferências e desvios. O foco deixa de ser no acompanhamento de cronograma e sim na execução do trabalho de cada dia.

O *Product Owner* possui vários problemas, seja a falta de tempo de um PO com o nível de poder adequado, ou a falta de poder de um PO alocado com tempo suficiente para o time sendo adicionado ainda as visões controversas dos vários *Stakeholders* que em alguns casos levam a prática de grupos de *Product Owner*.

Cohn (2010) apresenta como solução para o excesso de trabalho do PO um time de *Product Owner*, como relatado por Croix e Easton (2008) no estudo de caso em uma empresa britânica com desenvolvimento interno em apenas um projeto. Esta estratégia concorda com a proposta de Vlaanderen et al (2011) na definição da *Sprint* de Gerenciamento de Produtos realizada pelo Grupo de Gerentes do Produto. Sverrisdottir; Ingason e Jonasson (2014) realizaram um estudo sobre a prática de aplicação do PO e constataram tanto a implementação de um Time de PO quanto a existência de um PO técnico, concluindo que nenhuma das estratégias era adequada pois quando haviam conflitos de interesse, principalmente na priorização das histórias técnicas e de negócio, o Time de Desenvolvimento ficava sem direcionamento.

Em um caso do Departamento de Defesa Americano Saddington (2012) reforça a necessidade de um PO por Time de desenvolvimento, mesmo quando se trata de um projeto com vários *Stakeholders* ou vários times de desenvolvimento.

Em caso similar no desenvolvimento para a BBC Lowery e Evans (2007) relatam a importância da agilidade nas decisões do PO que caso não sejam realizadas para o time de Desenvolvimento, ou pior o induzem a tomar as decisões por conta própria.

Honious e Clark (2006) apresentam o caso de adoção do *Scrum* na *Reed Elsevier* no qual utilizaram a estratégia “*learn by doing*” como forma de diminuir o tempo de treinamento. Na qual ao invés de gastar tempo em treinamento utilizam *coaches* ágeis para adoção do *Scrum*. A mesma estratégia é utilizada em uma empresa com outsourcing relatado por Roch e Vasquez-Mccall (2009) que adicionaram o uso bem sucedido de um *workshop* de um dia para o treinamento e ensaio do *Scrum* com o Time *Scrum* na abertura do projeto.

Similar a este contexto Sutherland et al. (2008) apresentam o *Scrum* de *Scrums* em um projeto de *offshoring*, utilizado para projetos grandes, com mais de um time e *Backlogs* do Produto interdependentes. Neste caso aplicaram reuniões de acompanhamento entre os times permitindo o mapeamento de dependências entre os projetos através de reuniões periódicas entre os POs de cada time, sendo neste caso definido um PO dos PO's para tomar decisões de integração entre o time, evitando assim a paralisia.

Ozawa e Zhang (2013) define um PO Proxy para resolver questões culturais entre o PO do Japão e o Time de Desenvolvimento chinês. Este PO Proxy foi introduzido para auxiliar na definição dos requisitos e na compreensão tanto da língua quanto da cultura japonesa, conectando os dois mundos tanto culturalmente quanto tecnologicamente. Mas as dúvidas a respeito dos requisitos ainda eram respondidas diretamente pelo PO japonês, significando que o PO *Proxy* não o substituí, apenas auxiliava a comunicação entre as duas culturas.

Zieris & Salinger (2013) relatam um caso de aplicação do *Scrum* em contexto de outsourcing em que ressaltam a importância do uso dos princípios do *Scrum* de inspeção e adaptação também em nível organizacional. Além disto implementam dois *Scrum Masters*, um no cliente e outro no fornecedor, utilizado como interface de comunicação entre os dois times. Sendo que o *Scrum Master* no Cliente não substitui a comunicação direta entre PO e o Time de desenvolvimento, mesmo que distantes. Hong et al. (2010) apresentam em um caso de customização do *Scrum* a criação do evento chamado *Master Sprint Planning* do qual participam apenas *Scrum Master* e *Product Owner* e outros especialistas para definir o *ROADMAP* de todo o projeto com os *milestones* e a quantidade de *Sprints*.

A Reunião Diária é considerada por Sutherland et al. (2007) em seu estudo de caso com outsourcing o maior responsável pelo sucesso da aplicação do *Scrum* e o mais importante dos eventos *Scrum*. Indicando uma relação de causa e efeito entre essas reuniões e o sucesso dos resultados do *Scrum*.

### 3.4. O SCRUM E A TRANSFERÊNCIA DE CONHECIMENTO

A partir das origens do *Scrum* com Takeuchi e Nonaka (1995) e da revisão dos estudos de caso apresentados percebeu-se estreita relação entre o Scrum a transferência de conhecimento. Esta seção apresenta o relacionamento entre o *Framework Scrum* com a Visão Sistêmica e os relaciona com o trabalho sobre transferência de conhecimento proposto por Brito (2013).

#### 3.4.1. CICLO SECI NO SCRUM PARA GOVERNO

Takeuchi e Nonaka (1995) apresentaram estudos específicos sobre a transferência de conhecimento nos quais definiram o ciclo SECI (Socialização - Externalização - Combinação - Internalização) para aprendizado organizacional apresentada na Figura 13 - Espiral SECI adaptado de Nonaka & Takeuchi 1997 (BRITO, 2013).

A externalização ocorre entre o PO e outros *Stakeholders* na definição do *Backlog* do Produto. O Time de Desenvolvimento externaliza seu conhecimento a respeito do sistema no desenho da solução e no planejamento para sua criação descrito no *Backlog* da *Sprint*.

A construção do Software no dia a dia pode variar entre os times a depender de seu processo (SCHWABER; SUTHERLAND, 2013), mas o *Scrum* deixa claro que deve existir socialização e externalização diárias, embora não defina quais as formas de externalização.

A construção do Incremento de Software pode ser caracterizada tanto como combinação de artefatos intermediários e temporários, quanto externalização direta a partir da socialização com o cliente.

A internalização de conhecimento a respeito do software ocorre em seu uso na reunião de revisão da *Sprint* e posteriormente com o sistema em produção.





**Figura 13 - Espiral SECI adaptado de Nonaka & Takeuchi 1997 (BRITO, 2013)**

Brito (2013) apresenta atividades de transferência de conhecimento num processo de gestão de demandas de desenvolvimento de software para um órgão público federal brasileiro (o mesmo órgão utilizado neste estudo), que se baseia, entre outras referências, nos estudos de Takeuchi e Nonaka (1997). A proposta de Brito (2013), baseada em modelos de transferência de conhecimento, não aborda a aprendizagem organizacional e a visão sistêmica.

### 3.4.2. VISÃO SISTÊMICA E APRENDIZAGEM ORGANIZACIONAL

Senge (2009) apresenta o pensamento sistêmico como ferramenta de avaliação dos problemas organizacionais que promove o aprendizado organizacional. Esse tipo de pensamento possui uma notação própria chamada de arquétipo de sistemas, sua execução se baseia em outras quatro disciplinas que são: Domínio Pessoal, Modelos Mentais, Visão Compartilhada e Aprendizagem Organizacional, todas elas colaboram entre si reforçando umas às outras.

Nesta pesquisa não foram encontrados relatos de combinação entre as disciplinas apresentadas por Senge (2009) em conjunto com o *Scrum*, embora o próprio *Scrum* as referencie de forma declarada. O Quadro 1 apresenta a relação entre as disciplinas para o aprendizado organizacional e o *Scrum*.

Disciplina	Definição (Peter Senge, 2009)	Disciplina no <i>Scrum</i>
Domínio Pessoal	Trata-se do controle sobre si mesmo para alcançar os objetivos estabelecidos e não se perder em resoluções de problemas. Exige comprometimento com a verdade, para romper com estruturas de auto engano.	Habilidades e maturidade do Time de Desenvolvimento que permitem o comprometimento
Modelo Mentais	Modelos de crenças que determinam o comportamento e tomada de decisão. É necessário aprender a externalizar para então controlar seus efeitos.	São úteis no dia a dia, discussões sobre design e principalmente na auto organização. Publicar os modelos mentais pode ser uma forma de questionar o que é necessário para a execução do incremento, eliminando desperdícios.
Visão compartilhada	É o propósito, o motivo que une as pessoas. Conhecido como objetivo comum tem o potencial para conduzir as pessoas a um mesmo objetivo	Claramente determinada na Meta da <i>Sprint</i> que orienta o trabalho de todos sempre. Também presente nos conceitos de <i>Ready</i> e <i>Done</i> e no princípio da Transparência
Aprendizagem organizacional	É a capacidade do grupo aprender e se comportar como um organismo coeso, pois a inteligência de um grupo não é a soma das inteligências individuais.	No dia a dia ao definir o que fazer, nas Reuniões de Revisão e Retrospectiva em que cada um utiliza as outras disciplinas para descrever a realidade e então definir um plano objetivo e possível de melhoria para o próximo ciclo com foco no time e não apenas em um indivíduo.
Visão Sistêmica	É a percepção da realidade através de sistemas de equilíbrio e de reforço. Combinando-os de forma a encontrar pontos de alavancagem para as mudanças desejadas.	Pode ser utilizada no entendimento do contexto (projeto, equipe e cliente) e do domínio (negócio e solução). Dominar esta habilidade pode possibilitar o time a ultrapassar obstáculos que sem ela nem se quer conseguiriam ser determinados.

**Quadro 6 – *Scrum* e as disciplinas para aprendizagem organizacional (Autor)**

Observa-se pela comparação apresentada que existe uma relação direta entre as disciplinas e as características dos atores e a postura exigida destes nos eventos *Scrum*. A utilização das disciplinas de aprendizagem organizacional por times *Scrum* carece de estudos mais profundos.

### 3.5. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Os empregos do *Scrum* apresentados associados a outros modelos relacionados a governança não o descaracterizam, mas, em alguns casos, põe em cheque a sua leveza e agilidade. Nota-se entre as adaptações implementadas que há uma tendência de se inserir *Sprints* adicionais para outras atividades como é o caso da *Sprint* de validação, o HIP ou a *Sprint* de gerenciamento de Produto que mantém o BP em estado de *Ready*.

Adaptações devem ser adotadas para o contexto e definidas no modelo de forma que não inviabilizem novas adaptações necessárias em tempo de execução. A capacidade de adaptar-se em tempo de execução depende da autonomia e capacidade das pessoas envolvidas no desenvolvimento de software, sendo este processo de adaptação entendido como o processo de software no contexto ágil.

Dada a importância da competência das pessoas no sucesso dos projetos ágeis as cinco disciplinas da aprendizagem organizacional apresentam uma agenda clara e evolutiva do domínio das capacidades que podem levar a uma execução bem sucedida do *Scrum*, principalmente a respeito da colaboratividade e da capacidade de tomar-decisões em tempo de execução. Estudos sobre essa recomendação e aplicação ainda são requisitados.

Fica claro o foco na conversação mesmo quando a distância é uma restrição do caso, sendo a reunião diária relatada como responsável pelo sucesso dos projetos que devem dar ao time a liberdade para gerenciar o seu trabalho sem supervisão gerencial do fornecedor, que deve prestar assistência ao time para remover impedimentos identificados. Deve-se confiar que o Time irá requerer renegociação da *Sprint* antes que esta venha a falhar.

Os casos apresentados estabelecem a importância de comunicação do PO com outros *Stakeholders* reforçando o definido no Guia *Scrum* a respeito do PO. Todos os autores concordam que em qualquer estratégia adotada deve existir alguém responsável pela última decisão e por assumir o risco.

No nível organizacional, acima do time de *Scrum*, o acompanhamento deve ocorrer apenas com base nos produtos entregues que podem ser apresentados antes do fim da *Sprint*. Ressalta-se a importância das pessoas e de um processo que se adapte a elas sendo a medida de seu sucesso o software produzido, não um processo prescritivo.

Com relação aos papéis do time *Scrum* o menos controverso é o Time de Desenvolvimento e o mais é o *Product Owner*. O *Scrum Master* é entendido como o gerente de projeto, todavia com uma mudança de postura e de conduta.

As adaptações aqui apresentadas não são um compêndio de todas as adaptações possíveis e servem apenas como inspiração para a criação de adaptações necessárias e pertinentes ao contexto estudado e aplicação ao estudo de caso.

.

---

## **CAPÍTULO 4 - MATERIAIS E MÉTODOS**

## 4.1. CONSIDERAÇÕES INICIAIS DO CAPÍTULO

Este Capítulo apresenta os matérias e métodos empregados neste trabalho. Inicia com a caracterização do estudo de caso através de análise documental e entrevistas com funcionários do órgão, descrevendo-o quanto a: estrutura organizacional; contingente de pessoas; as mudanças estruturais, contratos vigentes. E apresenta uma análise deste órgão com o *Framework* conceitual apresentado nos capítulos anteriores.

## 4.2. CARACTERIZAÇÃO DO ÓRGÃO

O órgão selecionado como estudo de caso é um Ministério do Governo Federal, especificamente a sua Coordenação-Geral de Tecnologia da Informação (CGTI), que responde a demandas internas de desenvolvimento de software. Estas demandas podem ser tanto departamentais com poucos usuários quanto demandas ministeriais e no âmbito federal, para todo o país, impactando centenas até milhares de usuários.

Este órgão passa por um momento de transformação evolutiva que se norteia pelo mesmo espírito declarado no manifesto ágil. Em seu Plano Diretor de Tecnologia da Informação (PDTI) para os anos de 2013 a 2015 (BRASIL, 2013) declara em sua missão valores como *transparência* e *colaboração*, ambos presentes no *Scrum*, sendo a transparência um de seus pilares, e a colaboração evidente em sua definição de *Time Scrum* (SCHWABER; SUTHERLAND, 2013)

Sua visão é "Ser reconhecida como parceira... bem como referência de Qualidade de Tecnologia da Informação na Administração Pública Federal" (BRASIL, 2013), o que demonstra seu propósito de buscar a melhor qualidade associada a uma visibilidade amigável e parceira aos seus clientes assim como prevê o Manifesto Ágil (BECK et al., 2001a).

No PDTI observa-se o diagnóstico do Ministério quanto a análise SWOT. As forças que colaboram para a mudança com ágeis são uma equipe comprometida, equipe aberta a mudanças de processos e práticas, equipe conhecedora do ambiente do MC, das práticas boas e ruins implementadas no passado e um ambiente saudável e colaborativo, todas estas características são favoráveis a mudança. Porém no contato com a equipe percebe-se que a equipe foi reticente a práticas e processos propostos.

Quanto as oportunidades de melhoria, a análise SWOT apresenta os normativos, acórdãos e solicitações de auditorias que sustentam ações de melhoria de processos e equipes

de TI, o patrocínio institucional para a implementação de mudanças nos serviços, estrutura, processos e pessoas da CGTI.

A agilidade da alta gestão do Ministério na tomada de decisões, que foi relatada como oportunidade de melhoria, obteve melhoria significativa com a publicação e uso da Metodologia de Gerenciamento de Projetos de Tecnologia da Informação como norma operacional no órgão, segundo os funcionários do próprio órgão.

Dentre as fraquezas apresentadas observam-se dois grupos: as que podem ser impactadas pelas propostas de adaptação do *Scrum* para um processo ágil de gestão de demandas de software, e para as propostas que se devem buscar alternativas de solução.

Dentre aquelas que podem sofrer impacto positivo direto da proposta tem-se:

1. Baixa qualificação dos servidores em processos de governança e gestão de serviços de TI que com processo incremental que propicia melhoria contínua pode sofrer alterações em tempo de execução sem grandes interrupções para cursos e treinamentos;
2. Metodologias e processos de trabalho não definidos e/ou formalizados: a proposta do emprego do *Scrum* é formalizada em um Processo Ágil de Gestão de Demandas de Desenvolvimento de Software (PAGDDS);
3. Conhecimento e processos críticos concentrados na equipe dos fornecedores: embora o conhecimento "como fazer" ("*Know how*") continue com o fornecedor o conhecimento do "O que" (o que é produto de software e o que é qualidade deste produto de software) passará a ser, de forma gradativa, controlado pela contratante através das interações contínuas, dos ciclos de melhoria e a transparência do processo e seus conceitos como *Ready* e *Done*.

Dentre as fraquezas que devem ser respeitadas têm-se: o quantitativo inadequado de servidores sendo sete a quantidade de funcionários de carreira atual e vinte e quatro a meta até o fim de 2015; a concentração em serviços terceirizados; o elevado volume de contratos a serem geridos e fiscalizados por um número insuficiente de servidores e; instalações físicas inadequadas (BRASIL, 2013).

A rotatividade das equipes de fornecedores, identificada como fraqueza neste órgão, foi solucionada pelo Supremo Tribunal Federal, que utiliza metodologia ágil baseada no *Framework Scrum*, através da métrica de Indicador de "rotatividade média de pessoal

(*turnover*) da célula *Scrum* durante a produção de um *Product Backlog*” associada aos acordos de níveis de serviço (BRASIL, 2012d) que desmotiva a alteração de mais de uma pessoa no time de desenvolvimento sem que haja penalidades contratuais sobre o efetivo recebido.

Segundo relato do Coordenador Geral de Tecnologia da Informação do órgão no I Workshop de Disseminação do Conhecimento CQTS (Centro de Qualidade e Testes de Software) realizado em 23 de janeiro de 2014 no auditório da FGA Gama, a arquitetura de tecnologia da informação heterogênea, identificada no PDTI como fraqueza, não será alterada para os sistemas em funcionamento, algo que deve ser tratado em manutenção, mas para os novos sistemas uma arquitetura padrão foi definida por outra frente de trabalho e está em vias de aprovação pelo órgão, devendo ser utilizada em novos projetos.

Dentre os contratos de serviços vigentes no órgão três possuem impacto no desenvolvimento de sistemas e foram definidos no mesmo edital. São eles o contrato de Desenvolvimento e Evolução de Sistemas, o de Sustentação do Ambiente de TIC e o de Apoio ao controle de qualidade, que engloba contagem de pontos de função. Todos envolvidos no processo de desenvolvimento de sistemas.

O processo de gestão de demandas de desenvolvimento de software está vigente no órgão descrito no Termo de Referência do Edital (BRASIL, 2011c) como Modelo de Desenvolvimento de Software e é baseado no *OpenUP* com pagamentos *percentis* para cada fase, sendo a métrica de tamanho funcional do sistema a Análise de Pontos de Função segundo o Manual do IFPUG (BRASIL, 2011d).

A comunicação neste contexto é formal, suas vias identificadas são Ordem de serviço, e-mails de comunicação, que servem como registro das requisições e as reuniões entre preposto e representantes do órgão.

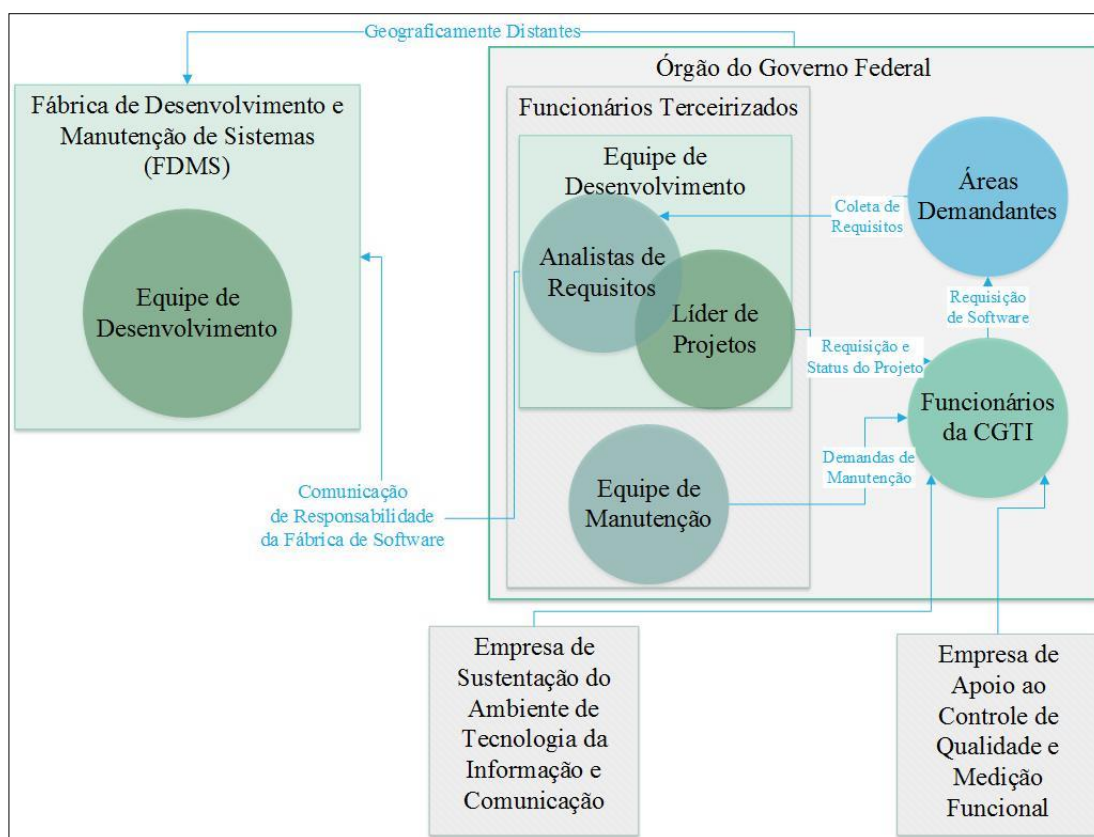
A Figura 14 – Contexto contratual do Estudo de Caso para os envolvidos no desenvolvimento de software, apresenta o esquema de alocação física dos envolvidos no desenvolvimento de software sendo estabelecidos no órgão apenas o Engenheiro de Requisitos e o Líder de Projetos da Fábrica de Desenvolvimento e Manutenção de Sistemas contratada, que podem ou não ser a mesma pessoa como apresenta o Diagrama de Veen e possuem comunicação direta tanto com a área demandante do órgão quanto com a CGTI e a equipe de desenvolvimento de novos sistemas fisicamente distante. A Equipe de



Manutenção responsável por pequenas manutenções encontra-se *in loco* mas não se relaciona com o Líder de Projetos.

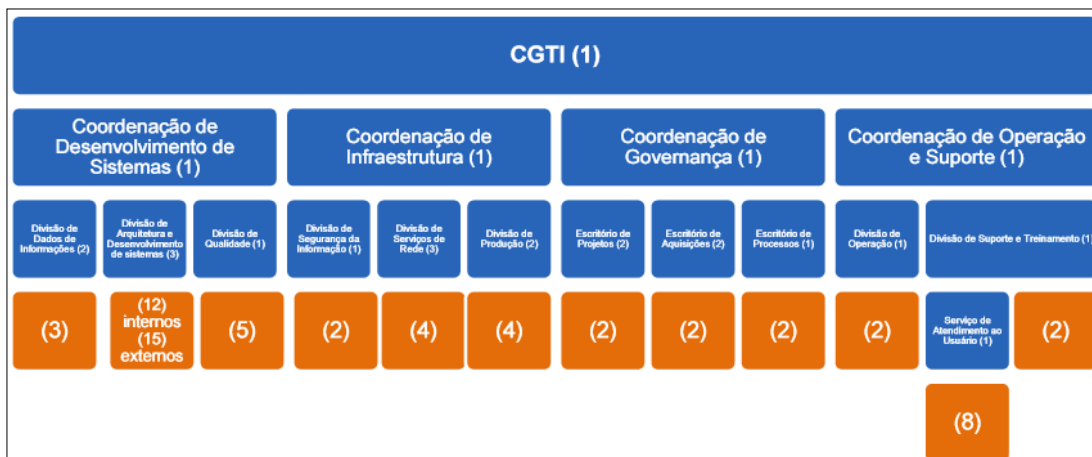
Na Figura 14 ainda apresenta as outras duas contratadas que podem ir a reuniões ou para resolver demandas pontuais, sendo que a contratada de infraestrutura possui funcionários *in loco*.

A comunicação direta entre equipe de desenvolvimento e cliente é considerada desnecessária na atual metodologia de desenvolvimento de software em que os requisitos são coletados pelo Engenheiro de Requisitos *in loco*.



**Figura 14 – Contexto contratual do Estudo de Caso para os envolvidos no desenvolvimento de software**

A Estrutura Organizacional da CGTI também tem sofrido alterações tendo sua última versão publicada em julho de 2012 apresentada na Figura 15 – Organograma CGTI do Estudo de Caso possui quatro coordenações distribuídas horizontalmente com média de três divisões/escritórios submetidos a estas. Dentre estas destaca-se a Coordenação de Desenvolvimento de Sistemas responsável por planejar, coordenar e acompanhar o desenvolvimento e a manutenção dos sistemas de informação do Ministério em sua arquitetura, modelo de dados e qualidade.



**Figura 15 – Organograma CGTI do Estudo de Caso**

Além da análise documental apresentada foram realizadas uma série reuniões em grupo e entrevistas com servidores do órgão no contexto do Projeto de Cooperação Técnica tanto para caracterização do contexto do órgão quanto da apresentação da proposta do Processo de Gestão de Demandas de Desenvolvimento de Sistemas.

Os clientes, pessoas da área demandante chamados no órgão de usuários chave, não possuem em sua maioria nenhum conhecimento técnico a respeito do desenvolvimento de software, além de terem uma agenda que inviabiliza a participação contínua no projeto.

O contrato atual possui um baixo valor do Ponto de Função para projetos de desenvolvimento, em comparação a outros órgãos. Enquanto as manutenções, são pagas em Horas de Serviço Trabalhadas (HST), mais vantajosas para a contratada.

Foram relatados problemas como: qualidade sendo aferida em documentos e não no código, baixa qualidade do software entregue, exigindo continuamente requisições de manutenção, atraso nas entregas, ausência de uma exigência clara dos tipos de teste exigidos para cada tipo de sistema, sendo requeridos em sua maioria apenas testes funcionais caixa preta.

A falta de sincronia entre os resultados da fábrica de desenvolvimento e de qualidade resultaram em problemas quanto ao tamanho do software desenvolvido, que foi aprovado para desenvolvimento com o dobro do tamanho validado.

### 4.3. GESTÃO DE PROJETO DO ÓRGÃO

A Metodologia de Gerenciamento de Projetos de Tecnologia da Informação (MGPTI) do órgão estudado foi promulgada no fim do ano de 2012, na forma de uma norma operacional. Esta norma define que todo projeto da Coordenação-Geral de Tecnologia da Informação (CGTI) deve ser executado segundo essa metodologia, desde projetos de aquisição de materiais a desenvolvimento de software, qualquer que seja a demanda executada pela TI do órgão.

A MGPTI possui sua base conceitual em práticas de gerenciamento de projetos consolidadas como Cobit 4.1 (PO 10), PMBoK - 4ª Edição e PRINCE2 e está em consonância com a MGP-TI do SISP/MP (BRASIL, 2011b). Foi definida com base em cinco premissas, que são: Documentação mínima, comunicação transparente, atribuições claras, processo flexível e informação documentada. Nota-se portanto que seus valores corroboram com os valores do manifesto ágil e com os pilares do *Scrum* (Transparência, inspeção e adaptação).

Seus objetivos são oito dos quais destacam-se: permitir o compartilhamento de informações e decisões sobre o projeto, respondendo a riscos identificados através de uma abordagem direta baseada em critérios, e possibilitar a gestão centralizada do portfólio de projetos de TI.

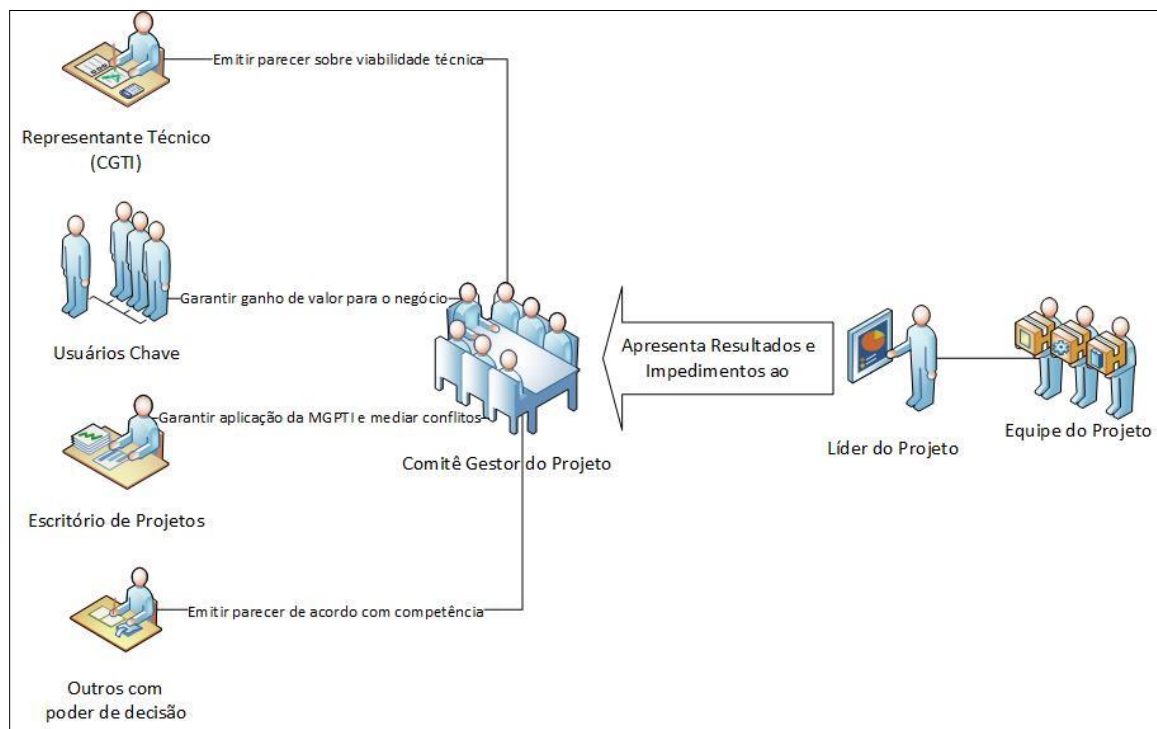
Esta metodologia é descrita em termos de papéis, fases, com lista de produtos definidos, e reuniões de decisão entre as fases.

Seu papel central é o Comitê Gestor do Projeto (CGP), responsável pelas decisões e suporte institucional ao projeto. Seu formato assemelha-se ao da Equipe de Alinhamento e Viabilidade da MGPTI do SISP/MP (BRASIL, 2011b), que por sua vez é idêntica a Equipe de Planejamento da Contratação definida na IN4 (BRASIL, 2010c).

A Figura 16 – Composição do CGP apresenta os integrantes do Comitê Gestor do Projeto, e seu relacionamento com os responsáveis pela produção da solução de TI. O Comitê Gestor do Projeto é composto por no mínimo três integrantes; o responsável técnico que emite parecer sobre a solução de TI apresentada; Usuários Chave com poder de decisão sobre o projeto que representam o negócio e o Escritório de Projeto que é responsável por disseminar a prática da MGPTI e garantir sua correta aplicação.

Numa analogia com o Time *Scrum* o Escritório de Projetos funciona como um *Scrum Master* para a MGPTI, os Usuários Chave com poder de decisão são como o PO, e nesse sentido deveria ser apenas um representante oficial dos usuários chave, como ocorre no *Scrum*, para evitar conflitos na tomada de decisão e o Responsável Técnico é como representante do Time de Desenvolvimento para a MGPTI. O responsável técnico é um integrante do órgão que critica a solução técnica apresentada e não faz parte da atual Equipe do Projeto (no *Scrum* seria o Time de Desenvolvimento) que só se relaciona com o Líder de Projeto.

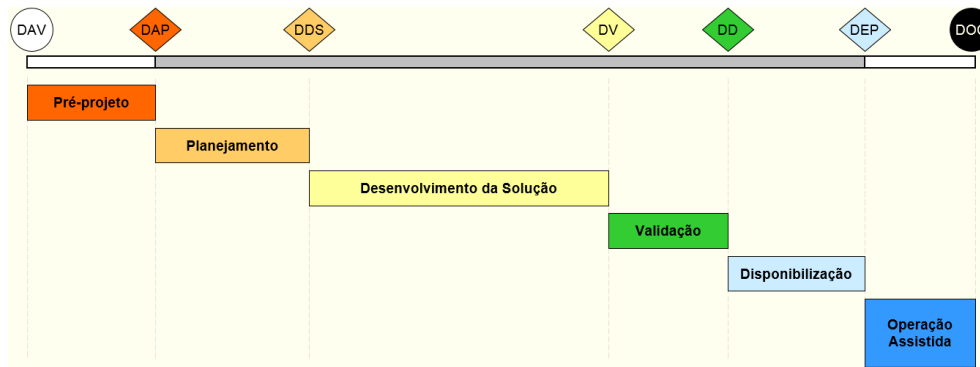
Este Líder de Projeto é o responsável direto pelo projeto e seus objetivos, possui a responsabilidade de reportar seu status e impedimentos ao Comitê Gestor do Projeto nas Reuniões de Decisão através de apresentações (Slides).



**Figura 16 – Composição do CGP (Fonte: Autor)**

Segundo a MGPTI todo projeto possui seis fases como mostra a Figura 17 – Fluxo da MGPTI (referência), os losangos entre as fases são reuniões de decisão em que participa todo o CGP, porém para as decisões de início e fim (DAV e DOC) não são convocadas reuniões e por isso seu símbolo é um círculo. Por não haver reunião de decisão na Decisão de Alinhamento e Viabilidade (DAV) o Comitê Gestor do Projeto só é definido na Decisão de Abertura do Projeto depois da Fase de Pré-Projeto.

Cada fase possui uma lista de produtos a serem apresentados e quais decisões devem ser tomadas. Em todas pode-se decidir continuar ou não com o projeto. Nestas reuniões de decisão também são apresentados os impedimentos e os produtos da fase anterior. Cada uma destas decisões possui um *template* de apresentação (*slides*) a ser utilizado pelo Líder de Projeto, responsável pela apresentação. A apresentação é mantida como documento de aprovação com assinatura dos integrantes do Comitê Gestor do Projeto.



**Figura 17 – Fluxo da MGPTI (referência)**

O Quadro 7 – Objetivos das Decisões apresenta os objetivos de cada uma das decisões que são executadas a nível de projeto, o que significa que mesmo o projeto como um todo mesmo que este seja dividido em *releases*.

Todo o processo se inicia quando a CGTI recebe do Coordenador Geral da Área Demandante um Documento de Oficialização da Demanda, no qual estabelece a necessidade do sistema e indica usuários chave, por sua vez a CGTI classifica a demanda como Projeto ou Operação Continuada. Para projetos realiza-se a DAV para demandas de operação continuada o fluxo segue para a DAP. No caso de ser novo projeto a CGTI designa equipe responsável pelo Pré-Projeto no qual se realiza o estudo sobre as alternativas de solução da demanda entre comprar, obter software livre e desenvolver e para tanto a Visão do Produto deve estar estabelecida. Somente na DAP o Comitê Gestor do Projeto é definido, assim como os outros papéis envolvidos.

Os produtos de cada fase são apresentados em três grupos: Objetivos, Solução e Gestão de Projeto porém o conjunto de produtos que os compõe em cada fase pode variar. Nota-se uma abordagem incremental para a produção de alguns produtos pelo uso dos termos preliminar, refinado e final entre parêntesis ao lado de vários produtos de todos os grupos.

Abreviação	Decisão	Objetivo
DAV	Decisão de Alinhamento e Viabilidade	Avalia o valor da demanda para o negócio e autorizar o início da fase de pré-projeto.
DAP	Decisão de Abertura do Projeto	Avalia o escopo e o planejamento preliminar e autorizar a abertura do projeto
DDS	Decisão de Desenvolvimento da Solução	Avalia o escopo, a solução apresentada e o planejamento final para o desenvolvimento da solução
DV	Decisão de Validação	Avalia se a solução está pronta para início de validação pelos usuários-chaves
DD	Decisão de Disponibilização	Avalia se a solução está pronta para disponibilização e se a organização está preparada para recebê-la
DEP	Decisão de encerramento do Projeto	Avalia a disponibilização da solução realizada, transfere a responsabilidade para a organização de suporte e manutenção e encerra o projeto
DOC	Decisão de Operação Continuada	Avalia a solução em operação em relação aos objetivos de negócio e, se necessário, identifica novas ações de melhoria

**Quadro 7 – Objetivos das Decisões**

Os produtos de Objetivos de Negócio podem ser inicialmente o DOD e depois Diagnóstico, Visão, Escopo e Limitações e as mudanças de escopo entre uma reunião de decisão e outra. Os produtos de Gerenciamento de Projeto são no mínimo o cronograma, o orçamento, a lista de riscos e as necessidades de recursos, podendo possuir também a lista de interessados, as dependências de outros projetos e por fim as lições aprendidas.

Dentre os três grupos, o que mais varia é o grupo de produtos da Solução definidos de acordo com a fase do projeto.

Dada a sua flexibilidade um mesmo projeto pode assumir mais de uma fase de planejamento, desenvolvimento, validação e implantação, permitindo assim sua adequação a um modelo incremental.

Segundo relato de funcionários do órgão as reuniões de decisão da MGPTI não poderiam ser convocadas mensalmente justamente pela composição do comitê gestor do projeto que reúne pessoas com poder de decisão em uma empresa com forte estrutura hierárquica, comunicação formal e agendas conflitantes.

## 4.4. CONSIDERAÇÕES FINAIS DO CAPÍTULO

A MGPTI embora possua sua base em modelos de gerência formais e bem estruturados como o PMBoK, a MGP-SISP e o COBIT 4.0 possui também o PRINCE2, que tem foco em resultados objetivos, e seus valores norteadores também vão de encontro as metodologias de desenvolvimento ágil de software e ao manifesto, com foco no produto resultante e sua qualidade efetiva.

Constata-se o alinhamento estratégico da organização aos valores ágeis, estabelecendo a construção de um ambiente que possa, na medida permitida a um órgão da Administração Pública Federal, tornar-se ágil.

Percebe-se grande similaridade entre a composição do Comitê Gestor do Projeto e o Time *Scrum* em suas funções como apresenta a comparação no Quadro 8 – Comparação entre papéis da MGPTI e *Scrum*, com a ressalva de que o representante técnico não representa o Time de Desenvolvimento, mas funciona como uma espécie de PO técnico que avalia a qualidade do Produto.

<b>Comitê Gestor do Projeto MGPTI</b>	<b>Time <i>Scrum</i></b>
Usuários Chave	<i>Product Owner</i>
Representante Técnico (CGTI)	Time de Desenvolvimento
Escritório de Projeto	<i>Scrum Master</i>

**Quadro 8 – Comparação entre papéis da MGPTI e *Scrum***

Porém essa similaridade não se configura como equivalência graças a diferença entre o nível de atuação: um atuando a nível de Time e outro a nível de Programa, sendo o Comitê de TI do órgão o representante no nível de portfólio e responsável pelo planejamento dos temas de investimento, descritos no Plano Diretor de TI, como definido no SAFe (LEFFINGWELL, 2011).

Neste caso, o Time de Desenvolvimento efetivo está isolado ao contato com o Líder de Projeto que nem é um papel existente no *Scrum* dado o conceito de Time *Scrum*, em que não existe liderança objetiva e todos devem ser capazes de relatar o status do projeto (SCHWABER; SUTHERLAND, 2013).



---

## **CAPÍTULO 5 - ADAPTAÇÕES DO *FRAMEWORK SCRUM* PARA UM ÓRGÃO PÚBLICO BRASILEIRO**

## 5.1. CONSIDERAÇÕES INICIAIS DO CAPÍTULO

Tendo apresentado o contexto de contratações públicas de Tecnologia da Informação no Brasil, as iniciativas de adoção de metodologias ágeis no governo brasileiro e em outros países, tendo declarado quais os empregos do *Scrum* e suas combinações com outras metodologias de gerenciamento de produto, neste Capítulo apresenta-se o Emprego do *Scrum* no Processo Gestão de Demandas de Desenvolvimento Ágil de Software (PAGDDS).

## 5.2. PROCESSO ÁGIL DE GESTÃO DE DEMANDAS DE DESENVOLVIMENTO DE SOFTWARE - PAGDDS

O propósito do PAGDDS é orientar o relacionamento entre contratante e contratada em um órgão público federal brasileiro nos moldes da lei. Seus princípios são os mesmos do *Scrum*, sobrescritos da seguinte forma:

Transparência entre o órgão e o fornecedor é plena, porém a transparência entre o fornecedor e o órgão é limitada pelo princípio ágil “*Software funcionando é a medida primária de progresso*” (BECK et al., 2001b), e em concordância com este, o padrão *Scrum* “Águas Passadas”, significando que órgão não terá visibilidade sobre o trabalho de outra forma se não por via de software funcional, descartando a necessidade do Gráfico *Burndown*.

Já a inspeção, ocorre em dois níveis. O primeiro nível é o do processo em si, nas atividades de avaliação da qualidade ao fim de cada subprocesso, quando é produzido um Relatório de Qualidade, com resultado objetivo, que autoriza a convocação das reuniões previstas pela MGPTI.

De forma complementar, a inspeção ocorre no nível do produto de software em três níveis distintos (Produto, *Release* e *Sprint* (LEFFINGWELL, 2011) (TURLEY, 2010)), através dos conceitos de *Ready* e *Done*, que permitirão o controle e acompanhamento da qualidade como recomenda a legislação (BRASIL, 1993), adaptando a proposta de Davis (2013) aos três níveis de controle definidos para este contexto e expandindo-a ao conceito de *Ready*, fornecendo assim um método estruturado para o alcance da excelência desejada pelo órgão (BRASIL, 2013).

A adaptação do processo além do que este permite é vetada pela caracterização do objeto contratual. Da mesma forma os critérios de qualidade e dos componentes do Incremento de Software não podem ser alterados (BRASIL, 2013). A iniciativa de alteração exige do órgão disponibilidade para negociar a evolução da qualidade esperada com o fornecedor.

A adaptação do *Backlog* do Produto é aceitável, desde que não fira os princípios da Administração Pública Federal (BRASIL, 1967), nem se altere o objeto da ordem de serviço estabelecido na Meta do Produto.

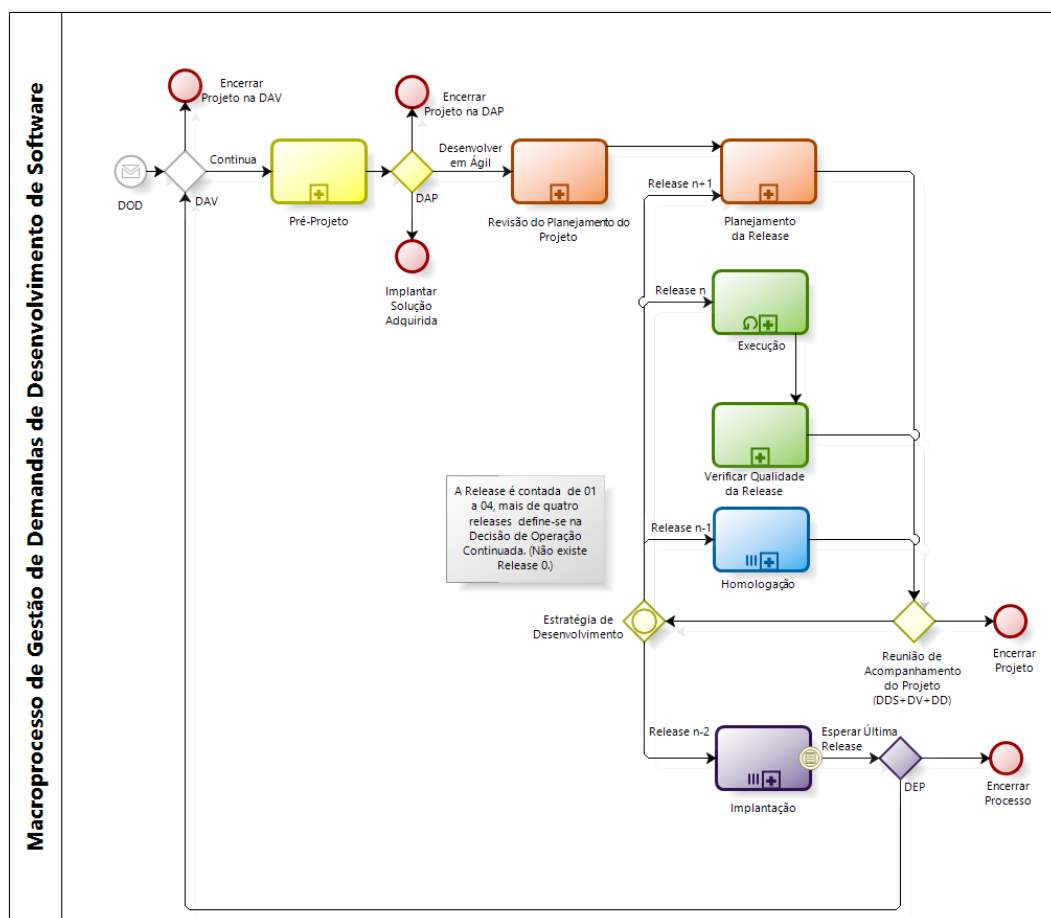
A adaptação do *Backlog* da *Sprint*, assim como no *Scrum* não é tolerada, ocorrendo apenas o esclarecimento de detalhes deste escopo.

A *Release* é a medida de produto para o Comitê Gestor do Projeto (CGP). Composta por uma a quatro *Sprints* que por sua vez variam de duas a quatro semanas como sugerido pelo Guia *Scrum* (SCHWABER; SUTHERLAND, 2013).

O tamanho mínimo dos releases será de duas semanas e no máximo 4 meses. Segundo as entrevistas com funcionários do órgão a média de tempo dos projetos no processo atual é de 8 meses com *releases* de no mínimo 4 meses, o que estabelece a igualdade em prazo do pior caso (mais demorado) do PAGDDS com a média do estado atual. O que permite uma introdução incremental do processo.

Embora a *Sprint* deva se manter com o mesmo tamanho ao longo do projeto (SCHWABER; SUTHERLAND, 2013) a mesma restrição não vale para a *release* que em um mesmo projeto pode variar a sua quantidade de *Sprints*.

A Figura 18 apresenta o modelo BPMN (*Business Process Model and Notation*) do PAGDDS em conformidade com a MGPTI do órgão. As cores representam fases do processo assim como o modelo *Stage-Gate* definido por Cooper (1990) em concordância com o PRINCE2 (TURLEY, 2010).



Powered by  
bizagi  
Modeler

**Figura 18 – Macroprocesso PAGDDS**

A fase de Planejamento da MGPTI foi dividida em dois subprocessos um para o Planejamento do Projeto, e outro para a *Release* de forma a permitir que se realize apenas o Planejamento da *Release* nos releases subsequentes de um mesmo projeto.

As reuniões de decisão da MGPTI (DAP, Reunião de Acompanhamento do Projeto e DEP) e DAV, que não é uma reunião, são representadas por *gates* exclusivos, em forma de losangos, nos quais executa-se uma decisão e apenas um caminho pode ser tomado (OMG, 2001).

A reunião de acompanhamento do projeto reúne as Decisões de Desenvolvimento de Sistema, de Validação e de Disponibilização permitindo assim sobreposição das fases de desenvolvimento das *Releases* como aconselham (SUTHERLAND, 2005) (TAKEUCHI; NONAKA, 1986) para as *Sprints*.

O *gate* “Estratégia de Desenvolvimento”, do tipo OU-INCLUSIVO significa que o fluxo poderá seguir a partir daí por um ou mais caminhos (OMG, 2001) dependendo do artefato “Estratégia de Desenvolvimento” que é o planejamento das decisões do CGP.

Assim como todos os eventos *Scrum* são *time-boxed*, todas as atividades do PAGDDS também são. O Quadro 9 – *Time-box* dos Subprocessos do PAGDDS apresenta o *time-box* dos subprocessos do PAGDDS sendo estes a soma do *time-box* das atividades que o compõe. Para as aproximações foram utilizados um dia de 8 horas de trabalho.

Fase da MGPTI	<i>Time-Box</i> PAGDDS
Pré-Projeto	15,5 horas Aproximadamente. 2 dias
Planejamento	Revisão do Planejamento do Projeto = 10hs Aproximadamente 1 dia e ¼ de dia
Execução	Depende do Planejamento. <i>Sprint</i> 2 a 4 semanas <i>Release</i> 1 a 4 <i>Sprints</i> Mínimo = 2 semanas Máximo = 4 meses
Homologação	Depende do planejamento.
Implantação	Depende do planejamento.

**Quadro 9 – *Time-box* dos Subprocessos do PAGDDS**

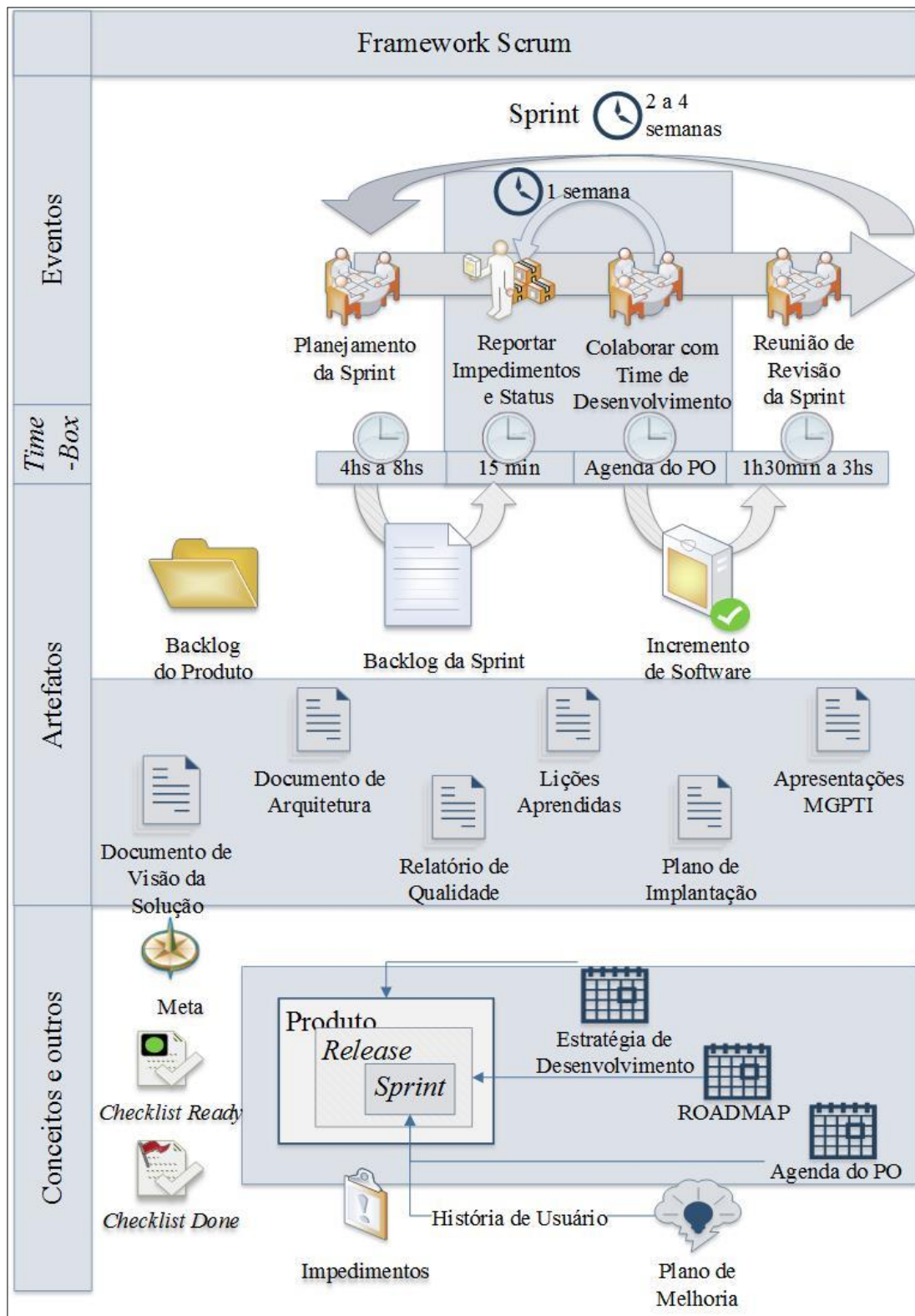
Os *time-boxes* estipulados para as atividades não consideram prazos para interrupções ou esperas e requerem a validação em pelo menos um projeto piloto.

O Quadro 10 – Estrutura do SAFe e Estudo de Caso relaciona a estrutura geral do *Scaled Agile Framework* (LEFFINGWELL, 2011) com o contexto do PAGDDS endereçando o PDTI no nível de portfólio para o estudo de caso da Coordenação-Geral de TI do Órgão.

Níveis do SAFe (LEFFINGWELL, 2011)	Estudo de Caso	Responsável
Portfólio	PDTI	Comitê de TI definido pela IN4
Programa	<i>ROADMAP</i> , Visão do Produto e <i>Backlog</i> do Produto.	Comitê Gestor do Projeto e <i>Product Owner</i>
Time	<i>Sprint Backlog</i>	Time <i>Scrum</i>

**Quadro 10 – Estrutura do SAFe e Estudo de Caso**

A Figura 19 – *Framework Scrum* Adaptado para o Estudo de Caso apresenta o *Framework Scrum* adaptado para este estudo de caso com base na Figura 5 do Capítulo 3 deste trabalho, no qual o fundo verde indica quais foram as adições definidas. O *Framework* será revisado da mesma forma apresentada pelo Guia *Scrum*, tendo já revisado os princípios, as seções seguintes revisarão os papéis, eventos e por fim os artefatos.



**Figura 19 – Framework Scrum Adaptado para o Estudo de Caso**

### 5.3. O TIME SCRUM E O POP

Os papéis do PAGDDS foram definidos de acordo com as recomendações e adaptações encontradas na bibliografia, o MGP-SISP em conformidade com a IN4, e a estrutura interna do estudo de caso. Esta seção limita-se a apresentar os papéis do *Scrum* com as adições do Comitê Gestor do Projeto e do *Product Owner Partner*.

O Comitê Gestor do Projeto como definido na MGPTI (BRASIL, 2012) é o responsável por realizar as decisões de projeto no nível organizacional sob a ótica da governança, decidindo sobre a sua continuidade ou não, liberação de recursos e resolução de impedimentos a nível “intra” e “inter” organizacionais.

O Time *Scrum* para o PAGDDS segue a recomendação do *Framework Scrum* (SCHWABER; SUTHERLAND, 2013). O PO é usuário chave da área de negócio com poder sobre projeto e única interface oficial entre o negócio e o Time de Desenvolvimento, definido na Decisão de Abertura do Projeto (DAP) dentre os Usuários-Chave estabelecidos no Documento de Oficialização da Demanda com tempo e conhecimento suficiente sobre o negócio para executar o papel.

O Time de Desenvolvimento pertence a uma contratada, que no contexto atual se situa em outro estado. Porém na proposta de adoção do *Scrum* a comunicação entre PO e o Time de Desenvolvimento passa a existir de forma direta através do uso dos meios de comunicação fornecidos, seja e-mail, telefone ou videoconferências.

O *Scrum Master*, para a MGPTI é o Líder de Projeto, e antes desta era reconhecido como Gestor do Projeto. No contexto atual trabalha na sede do órgão estudado e se comunica diretamente com o Time de Desenvolvimento.

Adiciona-se a estes o *Product Owner Partner* (*Parceiro do PO*), que como o nome sugere é o parceiro do PO, proposto para diminuir o conflito de interesses que possa haver entre o *Scrum Master* da contratada de desenvolvimento e do *PO*. Este papel serve ao PO como o *Scrum Master* deveria. A proposta deste papel é resultado de um ambiente em que o contrato não pode ser menos importante que a colaboração contrariando assim o valor ágil, como estabelecido por (BATRA, 2009) (BRASIL; 2013).

As responsabilidades do *Scrum Master* assumida pelo POP são:

- “Encontrar técnicas para o gerenciamento efetivo do Backlog do Produto;



- *Comunicar claramente a visão, objetivo e itens do Backlog do Produto para o Time de Desenvolvimento;*
- *Ensinar o Time Scrum a criar itens de Backlog do Produto de forma clara e concisa;*
- *Compreender a longo-prazo o planejamento do Produto no ambiente empírico;*
- *Compreender e praticar a agilidade; e,*
- *Facilitar os eventos Scrum conforme exigidos ou necessários.* “ (SCHWABER; SUTHERLAND, 2013)

Assemelha-se ao PO Proxy (OZAWA; ZHANG, 2013) pois deve diminuir a distância cultural entre o negócio e o time de desenvolvimento, não como interface de comunicação, mas como facilitador. Deve ser ainda capaz e auxiliar o PO na resolução de conflitos e escalonamento de decisões quando alguma decisão gerar impasse entre o PO e outros envolvidos. Possui também a função de garantir o comprometimento do PO realizando as atividades do PO em pares.

O PO pode delegar a ele parte de suas responsabilidades (SCHWABER; SUTHERLAND, 2013) como a manutenção do *Product Backlog* para o POP, mas ciente de que a responsabilidade, inclusive legal sobre os atos continua sendo dele. Não se espera que o POP venha a assumir o lugar do PO ou o represente em nenhuma de suas atividades e que assim como o *Scrum Master* o POP é um servo, e não cabe a ele definir nada sem o aval e consentimento do PO.

Este papel possui característica temporária a depender da maturidade do PO que pode a seu próprio ônus e tendo atuado como PO em outros projetos rejeitar o auxílio do POP.

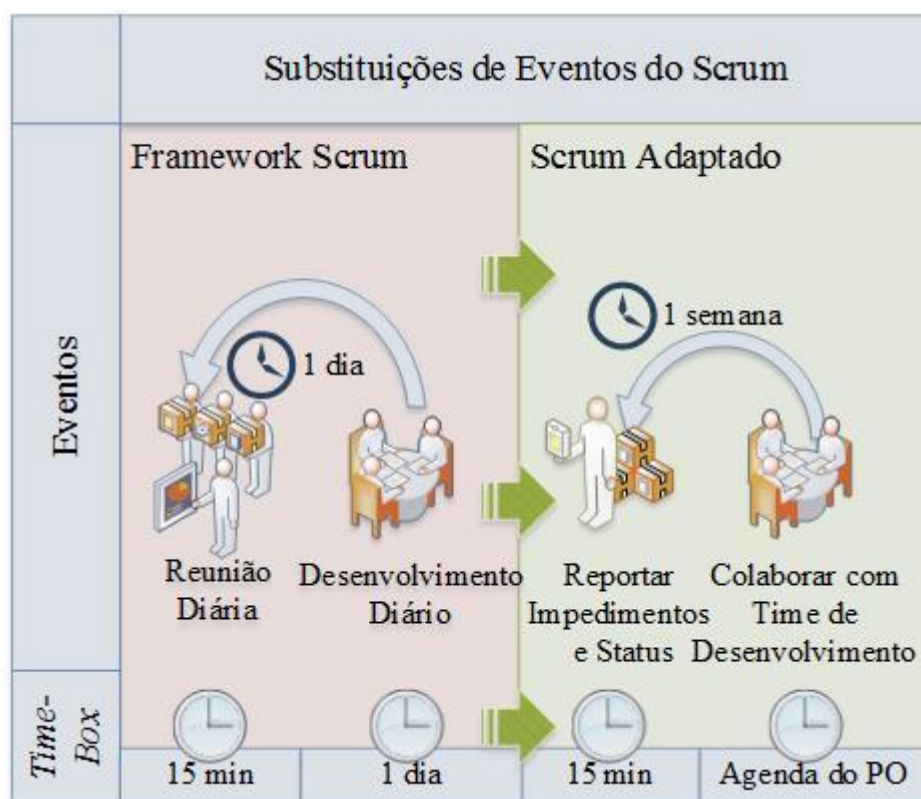
## 5.4. OS EVENTOS SCRUM NO PAGDDS

Esta seção apresenta as adaptações realizadas nos eventos do *Framework Scrum* apresentados na Figura 5 no Capítulo 3 para o este Estudo de Caso, todos estes eventos ocorrem por *Sprint* no subprocesso de Execução apresentado junto com os outros subprocessos no apêndice deste trabalho.

No contexto de um Processo de Gestão de Demandas criado a partir da ótica do contratante, não cabe a definição de eventos que sejam de responsabilidade exclusiva do Time de Desenvolvimento, pois não pode haver gerência ou interferência efetiva no trabalho realizado (BRASIL, 2003). Por esta razão somada a restrição de disponibilidade do PO, e ao

déficit declarado de funcionários da contratante propõe-se a alteração apresentada na Figura 20 – Eventos *Scrum* substituídos para o Estudo de Caso substituindo as atividades diárias por atividades em ciclos semanais.

A Reunião Diária que deve continuar sendo desenvolvida pelo Time de Desenvolvimento, é substituída pela atividade Reportar Impedimentos e *Status* da *Sprint* que é executada pelo *Scrum Master* na qual disponibiliza para o Escritório de Projetos, em uma conversa cara-a-cara, qual o status de execução da *Sprint* e relata os impedimentos que necessitam de sua intervenção.



**Figura 20 – Eventos *Scrum* substituídos para o Estudo de Caso**

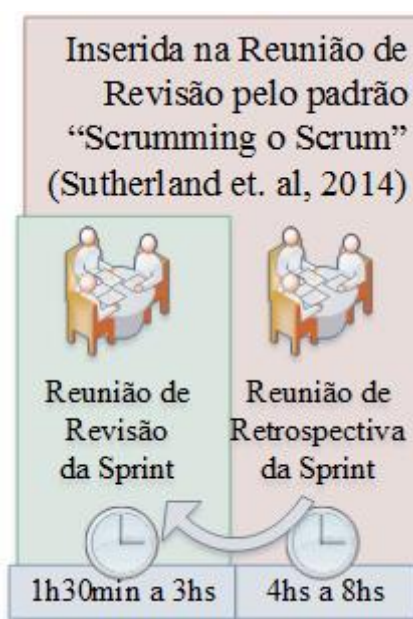
O reporte de *Status* visa definir um marco negocial e intermediário de avaliação da expectativa de sucesso da *Sprint*, sem a criação de nenhum artefato intermediário comprobatório, a apresentação do gráfico *Burndown* ou qualquer outra forma de monitoramento do status dependerá do *Scrum Master* e do Time de Desenvolvimento.

Já o reporte de impedimentos é prática recomendada pelo Guia *Scrum* e apresentada na Figura 10 – Fluxo de Trabalho do *Scrum* adaptado de (JAKOBSEN; SUTHERLAND, 2009), sendo o ciclo semanal o mínimo, podendo o *Scrum Master* relatar impedimentos sempre que forem identificados, desde que sejam da competência do órgão contratante. A

ausência ou falta de comprometimento do PO deve ser considerado um impedimento que deve ser relatado sempre que identificado.

A outra substituição é a do Desenvolvimento Diário que contaria com a participação constante do PO (SCHWABER; SUTHERLAND, 2013). Porém, por causa da restrição de tempo do PO e distância geográfica entre ele e o Time de Desenvolvimento propõe-se o evento Colaborar com o Time de Desenvolvimento que ocorrerá segundo disponibilidade do PO, ao menos uma vez por semana, preferencialmente no mesmo dia e horário e obrigatoriamente registrados na Agenda do PO. Nestes períodos ele estará disponível para responder a questões do Time de Desenvolvimento.

Para aplicação do *Scrum* no contexto do PAGDDS, para terceirização do desenvolvimento de software, é ainda proposta a supressão da Reunião de Retrospectiva da *Sprint*, que seria também de responsabilidade do Time de Desenvolvimento sem o envolvimento do PO. Todavia, considerando a melhoria no relacionamento entre PO e Time de Desenvolvimento ou a melhoria na qualidade das histórias de usuário propõe-se a integração da Retrospectiva na Reunião de Revisão da *Sprint* a partir do uso do padrão “*Scrumming o Scrum*” (SUTHERLAND; HARRISON; RIDDLE, 2014) como apresentado na Figura 21 – Inserção da Reunião de Retrospectiva da *Sprint* na Reunião de Revisão da . Considera-se para tanto a diminuição do tempo necessário afinal tratar-se-á o maior impedimento por vez e não se espera consumir mais do que 15 minutos do tempo estipulado para Reunião de Revisão.



### **Figura 21 – Inserção da Reunião de Retrospectiva da *Sprint* na Reunião de Revisão da *Sprint***

Embora exista similaridade no *Framework Scrum* no PAGDDS se propõe a atividade Escrever Histórias de Usuário que visa garantir o comprometimento do PO com a evolução do *Backlog* do Produto e relacionamento com o outros Stakeholders.

## **5.5. OS ARTEFATOS SCRUM NO PAGDDS**

A Figura 22 – Alteração apresenta a alteração dos artefatos entre o proposto no *Framework Scrum* apresentado na Figura 5 no Capítulo 3 para o Estudo de Caso.

Os três artefatos principais mantêm-se os mesmos (*Backlog* do Produto, *Backlog* da *Sprint* e Incremento de Software), quanto a categoria “Conceitos e outros” não tratados pelo Guia *Scrum* como artefatos, têm-se os impedimentos sem alterações e a definição de Metas e dos *checklists* de *Ready* e *Done* construídos e utilizados de forma incremental nos três níveis de controle estabelecidos para o escopo deste processo (Projeto, *Release* e *Sprint*).

Acrescenta-se nestes mesmos níveis de controle três tipos de cronograma de marcos. A Estratégia de Desenvolvimento define quais são os marcos previstos na MGPTI com reunião do Comitê Gestor do Projeto, o *ROADMAP* (LEFFINGWELL, 2011) relaciona as metas a suas datas de entrega, até o fim da Execução, não considerando homologação ou implantação e pôr fim a Agenda do PO que estabelece o comprometimento do PO com o projeto a cada *Sprint*.

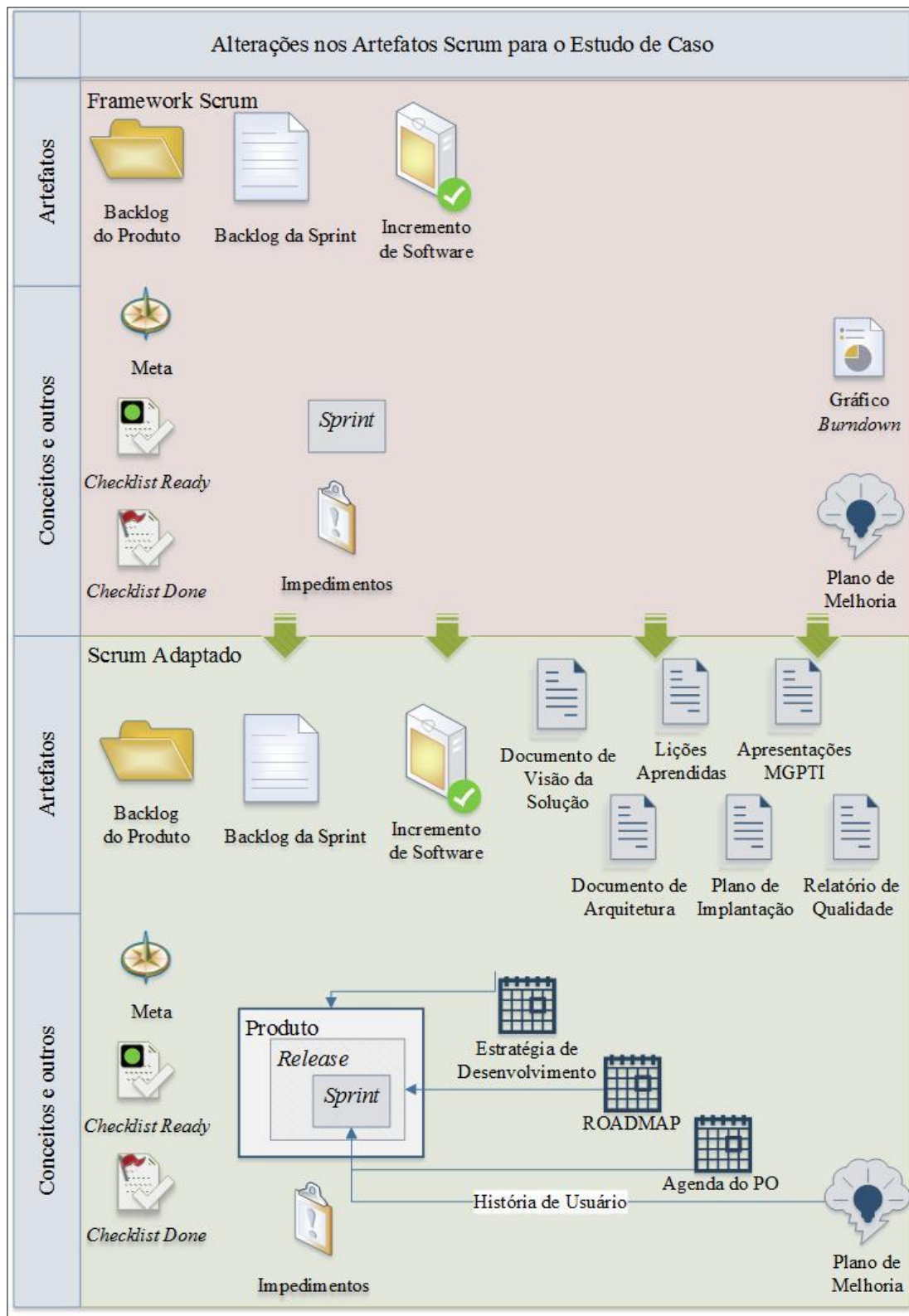
O Gráfico *Burndown* deixa de existir por ser considerado uma ferramenta do Time de Desenvolvimento com nenhum valor para o contratante, haja vista o princípio ágil de “*Software funcionando é a medida primária de progresso.*”(BECK et al., 2001b) em concordância com o valor ágil “*Software em funcionamento mais que documentação abrangente*”(BECK et al., 2001a) considerado pelo TCU como favorável ao princípio da eficiência desde que não o ferisse pela menos-valia da documentação (BRASIL, 2013).

São acrescentados ainda cinco artefatos necessários ao contexto e requeridos pela MGPTI. São eles o Documento de Visão da Solução, o Documento de Arquitetura, documento de Lições Aprendidas e as Apresentações da MGPTI.

Adicionalmente são propostos Relatórios de Qualidade para cada fase utilizados como “portas de qualidade” (JAKOBSEN; SUTHERLAND, 2009) nos subprocessos de forma

similar ao conceito de *Done* para o Incremento de Software, mas no nível de Projeto para cada fase/subprocesso.

O Plano de Melhoria não possui descrição no *Framework*, mas neste caso é compreendido como o *Kaizen* (História de Usuário de melhoria) com a utilização do padrão “*Scrumming o Scrum*” (SUTHERLAND; HARRISON; RIDDLE, 2014).



**Figura 22 – Alteração dos Artefatos *Scrum* para o Estudo de Caso**

---

## **CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS**

Este trabalho apresentou as adaptações no *Framework Scrum* para um estudo de caso de um Ministério do Governo Federal, podendo as propostas serem expandidas para outros casos de acordo com o seu contexto específico.

O *Scrum* a partir da perspectiva de um cliente governamental substitui todas as práticas do time de desenvolvimento a práticas orientadas ao PO e seu relacionamento com Time de Desenvolvimento. Adaptações do *Scrum* sob a ótica do contratante em terceirização ainda são escassas.

A existência da MGPTI baseada no PRINCE2 estabeleceu-se *a priori* como impeditiva, pois definia *gates* e fases estanques ao contrário do modelo “Tudo de uma vez” proposto pelo *Scrum*, mais similar ao caos. O estabelecimento de níveis de controle (Produto, *Release* e *Sprint*) foi a solução encontrada para organizar tanto a visibilidade do projeto, quanto a definição dos requisitos e acompanhamento da qualidade do produto.

A melhoria da qualidade possui um roteiro de evolução a ser seguido em cada projeto através das definições de *Ready* e *Done* nos três níveis de controle, orientando inclusive, o planejamento e execução do projeto, diminuindo assim a necessidade de outras *Sprints* de integração.

Embora espere-se que o órgão em sua adoção do *Scrum* passe pelos três níveis definidos por Sutherland (2005) e que os primeiros projetos se pareçam mais com o desenvolvimento atual com *Sprints* de integração, ao passo que a organização ganhe prática e adquira competência e possa sobrepor as atividades.

Dessa forma o próprio *Scrum* fornece um roteiro para a melhoria do processo a partir dos planos de melhoria que só poderá de fato ser executado com a melhoria da competência das pessoas que o executam.

Constata-se que para que o desenvolvimento ágil possa ocorrer em uma organização, esta precisa tornar-se ágil em seus processos internos para resolver impedimentos que surjam e seja capaz de receber os produtos o mais rápido possível.

O contato direto entre o PO e o cliente foi defendido a todo custo, pois entende-se a conversação como parte dos artefatos que não precisarão ser armazenadas, pois serão descritas no produto de software e não em artefatos intermediários.

Considera-se a agenda do PO um artefato importante no controle a respeito dos requisitos que foram definidos em histórias e testes de aceitação, pois fornece insumo para a comprovação do envolvimento do PO com outros *Stakeholders*, justificando possíveis descartes de *Sprints*.

Como trabalhos futuros seria de grande relevância a criação de definições de *Done* e *Ready* nos três níveis definidos neste trabalho em execução nos órgãos públicos, identificando suas similaridades e propondo um padrão destes *checklists* que pudessem ser compartilhadas no poder público através do SISP.



Um estudo de viabilização da proposta de *Code Camp* apresentada na seção 3.4.1 para contratação pública no Brasil em substituição ao processo licitatório.

Outro trabalho futuro deve-se a efetividade destas adaptações no contexto apresentado, validando assim as adaptações com um projeto piloto.

Conclui-se que embora adaptações possam e devam ser propostas de acordo com o contexto de aplicação, os melhores resultados se encontram no respeito ao exposto no Manifesto para o Desenvolvimento Ágil de Software, que estabelece uma conduta a ser tomada pelas pessoas que não pode ser traduzida em um processo.

Considera-se o *Scaled Agile Framework* (SAFe) um substituto natural do *Scrum* como base para a definição dos processos de gestão de demandas dos órgãos públicos federais brasileiros, pois fornece maior detalhamento que o *Scrum* e possui níveis de controle definição dos requisitos que concordam com os princípios da administração pública de planejamento e controle.

---

## REFERÊNCIAS BIBLIOGRÁFICAS

- AKER, S. et al. **Lessons learned and challenges of developing the NATO air command and control information services** Systems Conference (SysCon), 2013 IEEE International. **Anais...** In: SYSTEMS CONFERENCE (SYSCON), 2013 IEEE INTERNATIONAL. abr. 2013
- ANDRÉ, M. E. D. A. **Estudo de Caso em Pesquisa e Avaliação Educacional**. Brasília: Liber Livro Editora, 2005.
- BATRA, D. Modified Agile Practices for Outsourced Software Projects. **Commun. ACM**, v. 52, n. 9, p. 143–148, set. 2009.
- BECK, K. et al. **Manifesto para Desenvolvimento Ágil de Software**. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 10 jul. 2013.
- BEEDLE, M. et al. SCRUM: An extension pattern language for hyperproductive software development. **Pattern Languages of Program Design**, v. 4, p. 637–651, 1995.
- BOEHM, B.; TURNER, R. **Observations on balancing discipline and agility** Agile Development Conference, 2003. ADC 2003. Proceedings of the. **Anais...** In: AGILE DEVELOPMENT CONFERENCE, 2003. ADC 2003. PROCEEDINGS OF THE. 2003
- BRASIL. **Decreto-Lei nº 200, de 25 de fevereiro de 1967**. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/decreto-lei/del0200.htm](http://www.planalto.gov.br/ccivil_03/decreto-lei/del0200.htm)>. Acesso em: 4 nov. 2013.
- \_\_\_\_\_. **Lei nº 8.666, de 21 de junho de 1993**. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/l8666cons.htm](http://www.planalto.gov.br/ccivil_03/leis/l8666cons.htm)>. Acesso em: 4 nov. 2013.
- \_\_\_\_\_. **Decreto nº 2.271, de julho de 1997**. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/decreto/d2271.htm](http://www.planalto.gov.br/ccivil_03/decreto/d2271.htm)>. Acesso em: 4 nov. 2013.
- \_\_\_\_\_. **Lei 10.520 de 17 de julho de 2002**. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/2002/l10520.htm](http://www.planalto.gov.br/ccivil_03/leis/2002/l10520.htm)>. Acesso em: 3 maio. 2014.
- \_\_\_\_\_.; Tribunal Superior do Trabalho. **Súmulas 0331 a 0360**. Disponível em: <[http://www.dji.com.br/normas\\_inferiores/enunciado\\_tst/tst\\_0331a0360.htm](http://www.dji.com.br/normas_inferiores/enunciado_tst/tst_0331a0360.htm)>. Acesso em: 18 nov. 2013.
- \_\_\_\_\_.; Senado Federal; Tribunal de Contas da União. **Licitações & contratos: orientações e jurisprudência do TCU**. 4. ed. Brasília: Senado & Tribunal de Contas da União, 2010.
- \_\_\_\_\_. Ministério do Planejamento, Orçamento e Gestão; Secretaria de Logística e Tecnologia da Informação. **Instrução Normativa MP/SLTI Nº04**. Disponível em: <<http://www.governoeletronico.gov.br/sisp-conteudo/nucleo-de-contratacoes-de-ti/modelo-de-contratacoes-normativos-e-documentos-de-referencia/instrucao-normativa-mp-slti-no04>>. Acesso em: 23 abr. 2013c.
- \_\_\_\_\_.; \_\_\_\_\_.; \_\_\_\_\_. **Guia de Boas Práticas em Contratação de Soluções de TI**. Brasília: MPOG, 2011.

\_\_\_\_\_; \_\_\_\_\_. **Metodologia de Gerenciamento de Projetos do SISP (MGP-SISP)**. 1. ed. Brasília: MP, 2011b.

\_\_\_\_\_; Ministério das Comunicações. **ANEXO II - MODELO DE DESENVOLVIMENTO DE SOFTWARE - EDITAL DE PREGÃO ELETRÔNICO N.º 038/2011-MCSECRETARIA-EXECUTIVA SUBSECRETARIA DE PLANEJAMENTO, ORÇAMENTO E ADMINISTRAÇÃO**, , 12 jun. 2011c.

\_\_\_\_\_; \_\_\_\_\_. **ANEXO I - TERMO DE REFERÊNCIA - EDITAL DE PREGÃO ELETRÔNICO - N.º 038/2011-MCSECRETARIA-EXECUTIVA SUBSECRETARIA DE PLANEJAMENTO, ORÇAMENTO E ADMINISTRAÇÃO**, , 12 jun. 2011d.

\_\_\_\_\_; \_\_\_\_\_. Subsecretaria de Planejamento, Orçamento e Administração. **Norma Operacional SPOA No006**, 10 set. 2012.

\_\_\_\_\_. **SISP — Programa de Governo Eletrônico Brasileiro - Site Oficial**. Disponível em: <<http://www.governoeletronico.gov.br/sisp-conteudo>>. Acesso em: 27 abr. 2014a.

\_\_\_\_\_. **SLTI - Ministério do Planejamento, Orçamento e Gestão**. Disponível em: <<http://www.planejamento.gov.br/ministerio.asp?index=7&ler=s832>>. Acesso em: 27 abr. 2014b.

\_\_\_\_\_. Tribunal de Contas da União. **Guia de boas práticas em contratação de soluções de tecnologia da informação: riscos e controles para o planejamento da contratação**. Versão 1.0 ed. Brasília: BRASIL, 2012c.

\_\_\_\_\_. Supremo Tribunal Federal. **Edital PREGÃO ELETRÔNICO nº 84/2012**, 2012d. Disponível em: <<http://www.stf.jus.br/portal/edital/fazerDownload.asp?licitacao=22500&andamento=27904>>. Acesso em: 2 set. 2013

\_\_\_\_\_. Ministério do Planejamento. **Metodologia de Gerenciamento de Portfólio de Projetos do SISP (MGPP-SISP)**. 1. ed. Brasília: MP, 2013.

\_\_\_\_\_. Tribunal de Contas da União. **Portal do Tribunal de Contas da União - Fiscalização de Tecnologia da Informação**. Disponível em: <[http://portal2.tcu.gov.br/portal/page/portal/TCU/comunidades/tecnologia\\_informacao](http://portal2.tcu.gov.br/portal/page/portal/TCU/comunidades/tecnologia_informacao)>. Acesso em: 5 maio. 2014.

\_\_\_\_\_; \_\_\_\_\_. **Acórdão 2314/2013**. Disponível em: <<https://contas.tcu.gov.br/juris/SvlHighLight?key=ACORDAO-LEGADO-116247&texto=2b434f4c45474941444f253341253232504c454e4152494f2532322b414e442b2b2532384e554d41434f5244414f253341323331342b4f522b4e554d52454c4143414f253341323331342532392b414e442b2b2532384e554d414e4f41434f5244414f253341323031332b4f522b4e554d414e4f52454c4143414f25334132303133253239&sort=DTRELEVANCIA&ordem=DESC&bases=ACORDAO-LEGADO;DECISAO-LEGADO;RELACAO-LEGADO;ACORDAO-RELACAO-LEGADO;&highlight=&posicaoDocumento=0>>. Acesso em: 20 ago. 2013.

- \_\_\_\_\_.;.. **Acórdão 188/2010 - Plenário**. Disponível em:  
<<http://contas.tcu.gov.br/portaltextual/MostraDocumento?lnk=%28AC-0188-04/10-P%29%5bnumd%5d%5bB001,B002,B012%5d>>. Acesso em: 3 maio. 2014b.
- \_\_\_\_\_.; \_\_\_\_\_. **Acórdão 1287/2008 - Plenário**. Disponível em:  
<<http://contas.tcu.gov.br/portaltextual/ServletTcuProxy>>. Acesso em: 3 maio. 2014.
- COALLIER, F. Standards, agility, and engineering. **Computer**, v. 40, n. 9, p. 100–102, 2007.
- COHN, M. **Succeeding with agile: software development using Scrum**. Upper Saddle River, NJ: Addison-Wesley, 2010.
- COHN, M. L.; SIM, S. E.; LEE, C. P. What Counts as Software Process? Negotiating the Boundary of Software Work Through Artifacts and Conversation. **Computer Supported Cooperative Work (CSCW)**, v. 18, n. 5-6, p. 401–443, 1 dez. 2009.
- COOPER, R. G. Stage-gate systems: a new tool for managing new products. **Business horizons**, v. 33, n. 3, p. 44–54, 1990.
- COURSE. **What is PRINCE2?** Text. Disponível em: <<http://www.prince2.com/what-is-prince2>>. Acesso em: 13 jun. 2014.
- CROIX, A. DE S.; EASTON, A. **The Product Owner Team**IEEE, 2008Disponível em:  
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4599490>>. Acesso em: 14 fev. 2014
- CRUZ, C. S. **Governança de TI e Conformidade Legal no Setor Público: Um Quadro Referencial Normativo para a Contratação de Serviços de TI**. Brasília: Universidade Católica de Brasília, 2008.
- \_\_\_\_\_.; ANDRADE, E. L. P. DE; FIGUEIREDO, R. M. DA C. **Processo de Contratação de Serviços de Tecnologia da Informação para Organizações Públicas**. [s.l: s.n.].
- DAVIS, N. **Driving Quality Improvement and Reducing Technical Debt with the Definition of Done**IEEE, ago. 2013Disponível em:  
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6612893>>. Acesso em: 31 jan. 2014
- DINGSØYR, T. et al. A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, v. 85, n. 6, p. 1213–1221, jun. 2012.
- \_\_\_\_\_.; DYBÅ, T.; ABRAHAMSSON, P. **A Preliminary Roadmap for Empirical Research on Agile Software Development**IEEE, 2008Disponível em:  
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4599455>>. Acesso em: 27 fev. 2014
- DOWNEY, S.; SUTHERLAND, J. **Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft**2013 46th Hawaii International Conference on System

- Sciences (HICSS). **Anais...** In: 2013 46TH HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS). jan. 2013
- EUA - ESTADOS UNIDOS DA AMÉRICA; GAO - GOVERNMENT ACCOUNTABILITY OFFICE. **Software Development: Effective Practices and Federal Challenges in Applying Agile Methods**. Disponível em: <<http://www.gao.gov/assets/600/593091.pdf>>. Acesso em: 30 jan. 2014.
- FOWLER, M. **Using an Agile Software Process with Offshore Development**. Disponível em: <<http://martinfowler.com/articles/agileOffshore.html>>. Acesso em: 13 jun. 2014.
- GALLARDO-VALENCIA, R. E.; SIM, S. E. **Continuous and collaborative validation: a field study of requirements knowledge in agile** Managing Requirements Knowledge (MARK), 2009 Second International Workshop on. **Anais...** IEEE, 2009 Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5457339](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5457339)>. Acesso em: 13 fev. 2014
- GIL, A. C. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 2008.
- HAJJDIAB, H.; TALEB, A. S.; ALI, J. An Industrial Case Study for Scrum Adoption. **Journal of Software**, v. 7, n. 1, p. 237–242, 1 jan. 2012.
- HEEAGER, L. T.; NIELSEN, P. A. **Agile software development and its compatibility with a document-driven approach? A case study** Australasian conference on information systems, Melbourne, Australia. **Anais...** 2009 Disponível em: <<http://people.cs.aau.dk/~apr/ITV-MD3/Seminar1/HN09.pdf>>. Acesso em: 24 mar. 2014
- HONG, N.; YOO, J.; CHA, S. **Customization of Scrum Methodology for Outsourced E-Commerce Projects** Software Engineering Conference (APSEC), 2010 17th Asia Pacific. **Anais...** In: SOFTWARE ENGINEERING CONFERENCE (APSEC), 2010 17TH ASIA PACIFIC. nov. 2010
- HONIOUS, J.; CLARK, J. **Something to believe in [Reed Elsevier]** Agile Conference, 2006. **Anais...** IEEE, 2006 Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1667581](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667581)>. Acesso em: 18 fev. 2014
- HUMMEL, M. **State-of-the-Art: A Systematic Literature Review on Agile Information Systems Development** 2014 47th Hawaii International Conference on System Sciences (HICSS). **Anais...** In: 2014 47TH HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS). jan. 2014
- IDICIONÁRIO, A. **Significado de isonomia**. Disponível em: <<http://aulete.uol.com.br/isonomia>>. Acesso em: 26 abr. 2014.
- INGLATERRA. **The OGC Gateway™ Process: A manager's checklist**. Londres: GAO, 2007.

- \_\_\_\_\_; NAO - NATIONAL AUDIT OFFICE. **Governance for Agile delivery** | **National Audit Office**. Disponível em: <<http://www.nao.org.uk/report/governance-for-agile-delivery-4/>>. Acesso em: 30 jan. 2014.
- ISO/IEC/IEEE. **ISO/IEC/IEEE 24765: 2010 Systems and software engineering — Vocabulary**. Suíça: ISO, 2010.
- JAKOBSEN, C. R.; SUTHERLAND, J. **Scrum and CMMI Going from Good to Great** Agile Conference, 2009. AGILE '09. **Anais...** In: AGILE CONFERENCE, 2009. AGILE '09. ago. 2009
- JAMIESON, D.; VINSEN, K.; CALLENDER, G. **Agile procurement: new acquisition approach to agile software development** Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference on. **Anais...** IEEE, 2005a Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1517751](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1517751)>. Acesso em: 21 mar. 2014
- \_\_\_\_\_; \_\_\_\_\_. **Agile procurement to support agile software development** 2005 3rd IEEE International Conference on Industrial Informatics, 2005. INDIN '05. **Anais...** In: 2005 3RD IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 2005. INDIN '05. ago. 2005b
- \_\_\_\_\_; \_\_\_\_\_. **Agile Procurement and Dynamic Value for Money to Facilitate Agile Software Projects** 32nd EUROMICRO Conference on Software Engineering and Advanced Applications, 2006. SEAA '06. **Anais...** In: 32ND EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS, 2006. SEAA '06. ago. 2006
- JANES, A. A.; SUCCI, G. **The Dark Side of Agile Software Development** Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software. **Anais...** Onward! '12. New York, NY, USA: ACM, 2012 Disponível em: <<http://doi.acm.org/10.1145/2384592.2384612>>. Acesso em: 11 abr. 2014
- KENT BECK et al. **Manifesto para Desenvolvimento Ágil de Software**. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 10 jul. 2013a.
- \_\_\_\_\_. **Princípios por trás do Manifesto Ágil**. Disponível em: <<http://agilemanifesto.org/iso/ptbr/principles.html>>. Acesso em: 10 jul. 2013b.
- KUMAR, G.; BHATIA, P. K. **Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies** 2014 Fourth International Conference on Advanced Computing Communication Technologies (ACCT). **Anais...** In: 2014 FOURTH INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING COMMUNICATION TECHNOLOGIES (ACCT). fev. 2014
- LEFFINGWELL, D. **Scaling software agility: best practices for large enterprises**. Upper Saddle River, NJ: Addison-Wesley, 2007.

- \_\_\_\_\_. **Agile software requirements: lean requirements practices for teams, programs, and the enterprise**. Boston: Pearson, 2011.
- \_\_\_\_\_. **Scaling Software Agility | Best Practices for Large Enterprises**, by Dean Leffingwell. Disponível em: <<http://scalingsoftwareagilityblog.com/>>. Acesso em: 9 out. 2013a.
- \_\_\_\_\_. **Scaled Agile Framework Big Picture**. Disponível em: <<http://www.scaledagileframework.com/>>. Acesso em: 3 fev. 2013b.
- \_\_\_\_\_. **Hardening-Innovation-Planning. Scaled Agile Framework**, 14 mar. 2013c.
- LOWERY, M.; EVANS, M. **Scaling product ownership** Agile Conference (AGILE), 2007. **Anais...** IEEE, 2007 Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4293617](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4293617)>. Acesso em: 31 jan. 2014
- MAURER, F.; MELNIK, G. **Agile Methods: Crossing the Chasm** 29th International Conference on Software Engineering - Companion, 2007. ICSE 2007 Companion. **Anais...** In: 29TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING - COMPANION, 2007. ICSE 2007 COMPANION. maio 2007
- MELO, C. DE O. et al. The evolution of agile software development in Brazil. **Journal of the Brazilian Computer Society**, v. 19, n. 4, p. 523–552, 1 nov. 2013.
- MINISTÉRIO DAS COMUNICAÇÕES; BRASIL. **Plano Estratégico de Tecnologia da Informação (PETI) e Plano Diretor de Tecnologia da Informação (PDTI) de 2013 a 2015 Versão 2.0**, dez. 2013. Disponível em: <[www.mc.gov.br/doc-crs/doc\\_download/1147-plano-diretor-de-tecnologia-da-informacao-do-ministerio-das-comunicacoes-pdti-para-o-periodo-2013-2015](http://www.mc.gov.br/doc-crs/doc_download/1147-plano-diretor-de-tecnologia-da-informacao-do-ministerio-das-comunicacoes-pdti-para-o-periodo-2013-2015)>. Acesso em: 10 mar. 2013
- MOE, C. E.; NEWMAN, M. **The Public Procurement of IS -- A Process View** IEEE, jan. 2014 Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6758870>>. Acesso em: 21 mar. 2014
- MUNDRA, A.; MISRA, S.; DHAWALE, C. A. **Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software** IEEE, jun. 2013 Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6681108>>. Acesso em: 10 abr. 2014
- OBJECT MANAGMENT GROUP OMG. **Business Process Model and Notation (BPMN)**, 3 jan. 2001. Disponível em: <<http://www.omg.org/spec/BPMN/20100501>>. Acesso em: 2 jul. 2013
- OZAWA, H.; ZHANG, L. **Adapting Agile Methodology to Overcome Social Differences in Project Members** IEEE, ago. 2013 Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6612882>>. Acesso em: 6 fev. 2014



PRIES-HEJE, L.; PRIES-HEJE, J.; DALGAARD, B. **Scrum Code Camps**IEEE, ago. 2013Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6612880>>. Acesso em: 10 maio. 2014

**Published Patterns**. Disponível em: <<https://sites.google.com/a/scrumplp.org/published-patterns/>>. Acesso em: 10 abr. 2014.

RAUTIAINEN, K.; LASSENIUS, C.; SULONEN, R. 4cc: A framework for managing software product development. **ENGINEERING MANAGEMENT JOURNAL-ROLLA-**, v. 14, n. 2, p. 27–32, 2002.

\_\_\_\_\_.; VUORNOS, L.; LASSENIUS, C. **An experience in combining flexibility and control in a small company's software product development process**Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on. **Anais...**IEEE, 2003aDisponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1237962](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1237962)>. Acesso em: 16 maio. 2014

\_\_\_\_\_.; \_\_\_\_\_.; \_\_\_\_\_. **An experience in combining flexibility and control in a small company's software product development process**Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on. **Anais...**IEEE, 2003bDisponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1237962](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1237962)>. Acesso em: 16 maio. 2014

ROCHE, G.; VASQUEZ-MCCALL, B. **The Amazing Team Race A Team Based Agile Adoption**Agile Conference, 2009. AGILE '09. **Anais...** In: AGILE CONFERENCE, 2009. AGILE '09. ago. 2009

SADDINGTON, P. **Scaling Agile Product Ownership through Team Alignment and Optimization: A Story of Epic Proportions**IEEE, ago. 2012Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6298102>>. Acesso em: 6 fev. 2014

SÁNCHEZ, P. D. DE M.; GAYA, C. G.; PERÉZ, M. Á. S. Standardized Models for Project Management Processes to Product Design. **Procedia Engineering**, The Manufacturing Engineering Society International Conference, MESIC 2013. v. 63, p. 193–199, 2013.

SCHWABER, K. **Agile Project Management with Scrum**. [s.l.] Microsoft Press, 2004.

\_\_\_\_\_.; SUTHERLAND, J. **Guia do Scrum**. [s.l.: s.n.].

SCHWALBE, K. **Managing a Project Using an Agile Approach and the PMBOK® Guide**Proceedings of the Information Systems Educators Conference ISSN. **Anais...**2012Disponível em: <<http://iwww.intropm.com/downloads/schwalbe-agilepm-isecon-2012.pdf>>. Acesso em: 11 abr. 2014

SCRUM.ORG. **Scrum Guide™ | Scrum.org - The home of Scrum**. Disponível em: <<https://www.scrum.org/Scrum-Guide>>. Acesso em: 23 abr. 2014.

**Silicon Valley CodeCamp.** Disponível em: <<http://www.siliconvalley-codecamp.com/>>. Acesso em: 10 maio. 2014.

SUTHERLAND, J. **Future of scrum: parallel pipelining of sprints in complex projects** Agile Conference, 2005. Proceedings. **Anais...** In: AGILE CONFERENCE, 2005. PROCEEDINGS. jul. 2005

SUTHERLAND, J. et al. **Distributed Scrum: Agile Project Management with Outsourced Development Teams** 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007. **Anais...** In: 40TH ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 2007. HICSS 2007. jan. 2007

SUTHERLAND, J. et al. **Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams** Agile, 2008. AGILE '08. Conference. **Anais...** In: AGILE, 2008. AGILE '08. CONFERENCE. ago. 2008

SUTHERLAND, J.; HARRISON, N.; RIDDLE, J. **Teams That Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams** IEEE, jan. 2014 Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6759182>>. Acesso em: 10 abr. 2014

SUTHERLAND, J.; JAKOBSEN, C. R.; JOHNSON, K. **Scrum and CMMI Level 5: The Magic Potion for Code Warriors** Hawaii International Conference on System Sciences, Proceedings of the 41st Annual. **Anais...** In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, PROCEEDINGS OF THE 41ST ANNUAL. jan. 2008a

SUTHERLAND, J.; JAKOBSEN, C. R.; JOHNSON, K. **Scrum and CMMI Level 5: The Magic Potion for Code Warriors** Hawaii International Conference on System Sciences, Proceedings of the 41st Annual. **Anais...** In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, PROCEEDINGS OF THE 41ST ANNUAL. jan. 2008b

SVERRISDOTTIR, H. S.; INGASON, H. T.; JONASSON, H. I. The Role of the Product Owner in Scrum-comparison between Theory and Practices. **Procedia - Social and Behavioral Sciences**, Selected papers from the 27th IPMA (International Project Management Association), World Congress, Dubrovnik, Croatia, 2013. v. 119, p. 257–267, 19 mar. 2014.

TAKEUCHI, H.; NONAKA, I. The new new product development game. **Harvard business review**, v. 64, n. 1, p. 137–146, 1986.

**The Institute for Government.** Disponível em: <<http://www.instituteforgovernment.org.uk/>>. Acesso em: 6 maio. 2014.

TURLEY, F. **The PRINCE2® Foundation Training Manual.** [s.l: s.n.].

VANHANEN, J.; ITKONEN, J.; SULONEN, P. **Improving the interface between business and product development using agile practices and the cycles of control**

**framework**Agile Development Conference, 2003. ADC 2003. Proceedings of the. **Anais...** In: AGILE DEVELOPMENT CONFERENCE, 2003. ADC 2003. PROCEEDINGS OF THE. 2003

VLAANDEREN, K. et al. The agile requirements refinery: Applying SCRUM principles to software product management. **Information and Software Technology**, v. 53, n. 1, p. 58–70, jan. 2011.

WILLIAMS, L. et al. **Scrum + Engineering Practices: Experiences of Three Microsoft Teams**2011 International Symposium on Empirical Software Engineering and Measurement (ESEM). **Anais...** In: 2011 INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING AND MEASUREMENT (ESEM). set. 2011

YI, L. **Manager as Scrum Master**IEEE, ago. 2011Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6005499>>. Acesso em: 16 maio. 2014

ZIERIS, F.; SALINGER, S. **Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practices**IEEE, ago. 2013Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6613079>>. Acesso em: 12 abr. 2014

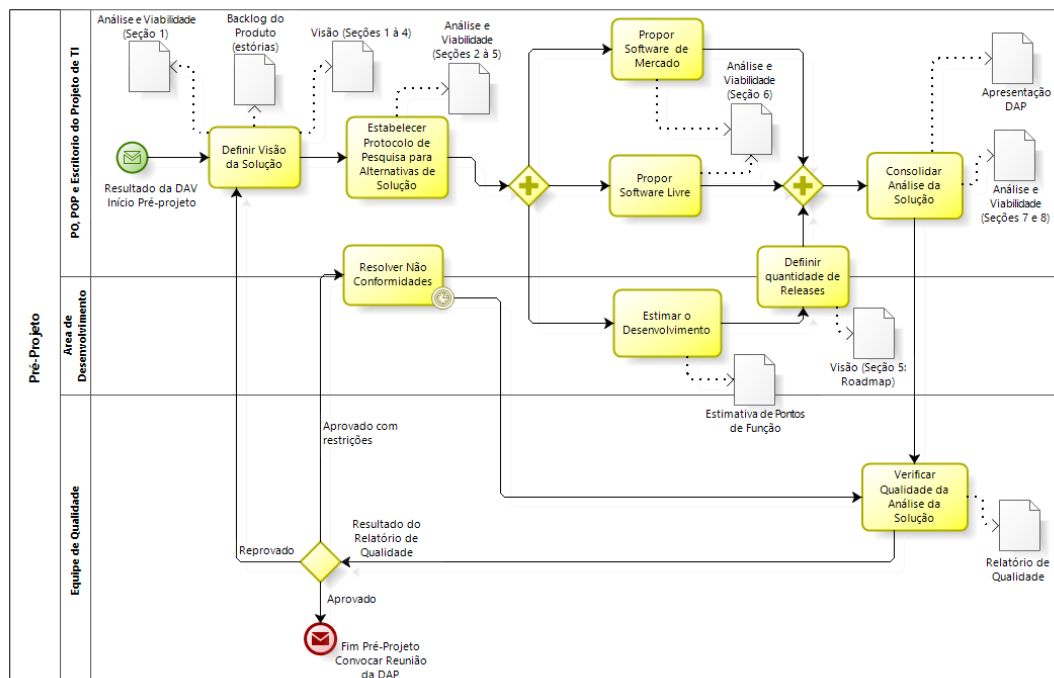
---

**APÊNDICE – APRESENTAÇÃO DOS SUBPROCESSOS DO  
PAGDDS**

Este apêndice apresenta os subprocessos do PAGDDS de acordo com o contexto apresentado e a bibliografia apresentada.

O PO é definido na DAV pelo Escritório de Projetos que é parte da Coordenação de Governança do estudo de caso e responsável por avaliar a demanda e dar continuidade a ela.

O Pré-Projeto é apresentado na Figura 23 – Subprocesso de Pré-Projeto do sendo ele a instanciação da Análise de Viabilidade da Contratação proposta pela IN4 (BRASIL, 2010c) no nível da contratação. Como neste caso a contratação foi realizada como serviço comum, esta análise não foi realizada para todas as demandas de software. O dono deste subprocesso é o Escritório de Projetos responsável por garantir o cumprimento dos prazos.

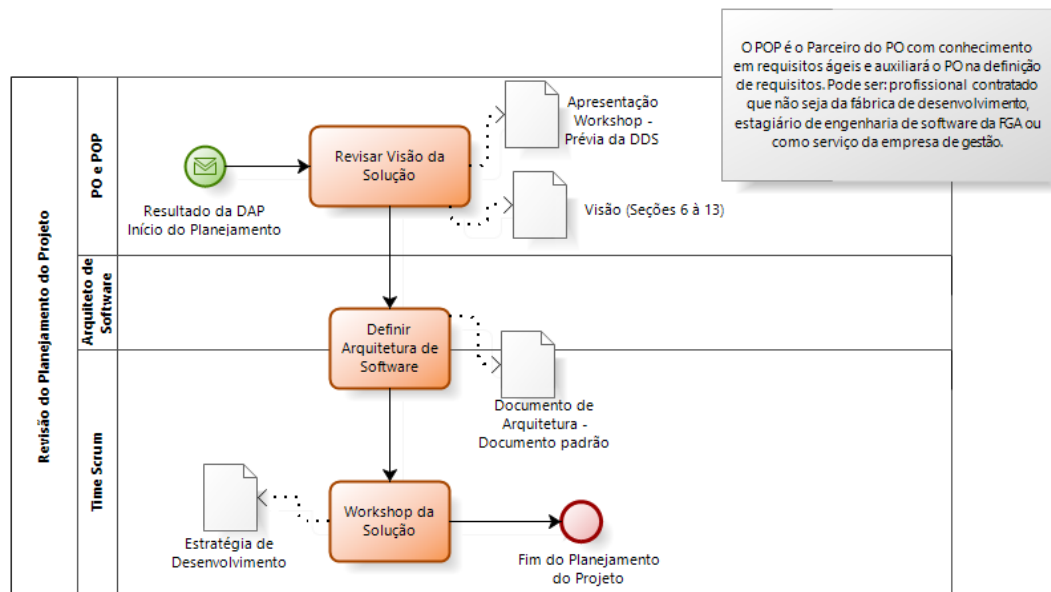


**Figura 23 – Subprocesso de Pré-Projeto do PAGDDS**

A primeira atividade “Definir Visão da Solução” é realizada pelo PO e o POP e um representante do Escritório de Projetos. Nela criam o Documento de Visão e o *Backlog* do Produto a partir do definido no Documento de Oficialização da Demanda.

No subprocesso “Revisão do Planejamento do Projeto” (Figura 24 – Subprocesso de Revisão do Planejamento do Projeto) a visão definida é revisada juntamente com o *Backlog*

do produto e prepara-se para o Workshop da Solução em que Time de Desenvolvimento é apresentado por videoconferência e passa a conhecer a demanda e o PO.

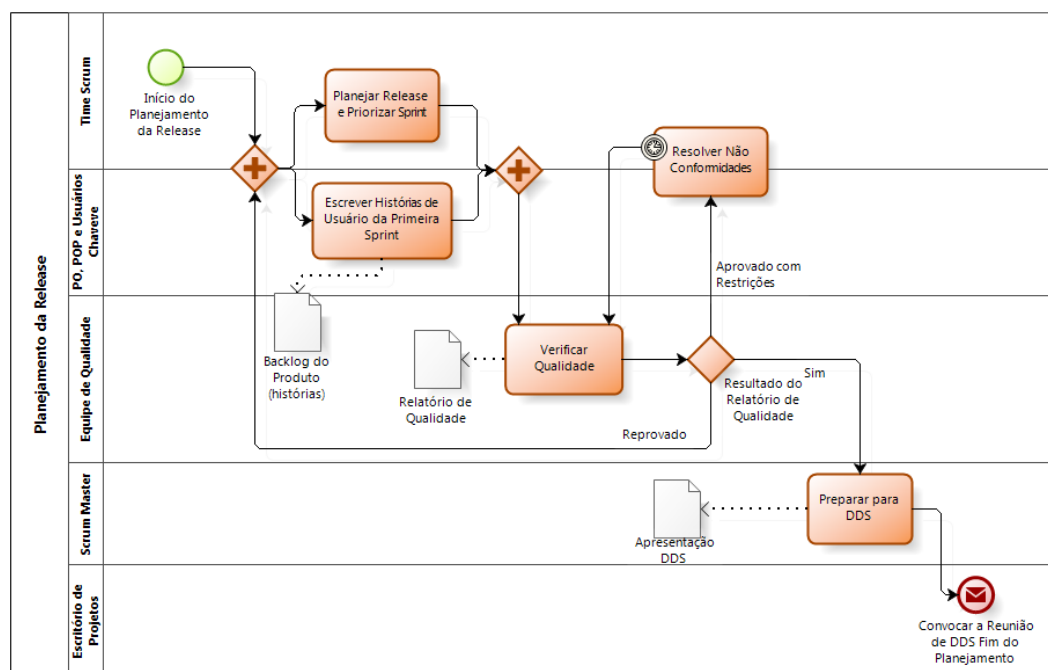


**Figura 24 – Subprocesso de Revisão do Planejamento do Projeto**

A atividade “Definir Arquitetura de Software” foi sugerida por funcionários do órgão para deixar claro um marco arquitetural, no entanto no *Scrum* mesmo a arquitetura do sistema é realizada de forma incremental, e no Workshop da Solução haverá apenas uma prévia desta de acordo com o padrão do órgão.

Ao fim do “Workshop da Solução” ocorrem tanto a definição da “Estratégia de Desenvolvimento” quanto o a fixação do tamanho da *Sprint* para o projeto, assim como a definição de *Ready* é acordada para os três níveis e a definição de *Done* nos níveis do Produto e da *Release*. Estes artefatos estão definidos na seção seguinte.

Os objetivos do Planejamento da *Release* (Figura 25 – Subprocesso de Planejamento da *Release* no PAGDDS) são planejar as *Sprints* do primeiro release e deixar o *Backlog* do Produto *Ready* para a primeira *Sprint*. A atividade escrever Histórias de Usuário da Primeira *Sprint*, é planejada na agenda do PO.



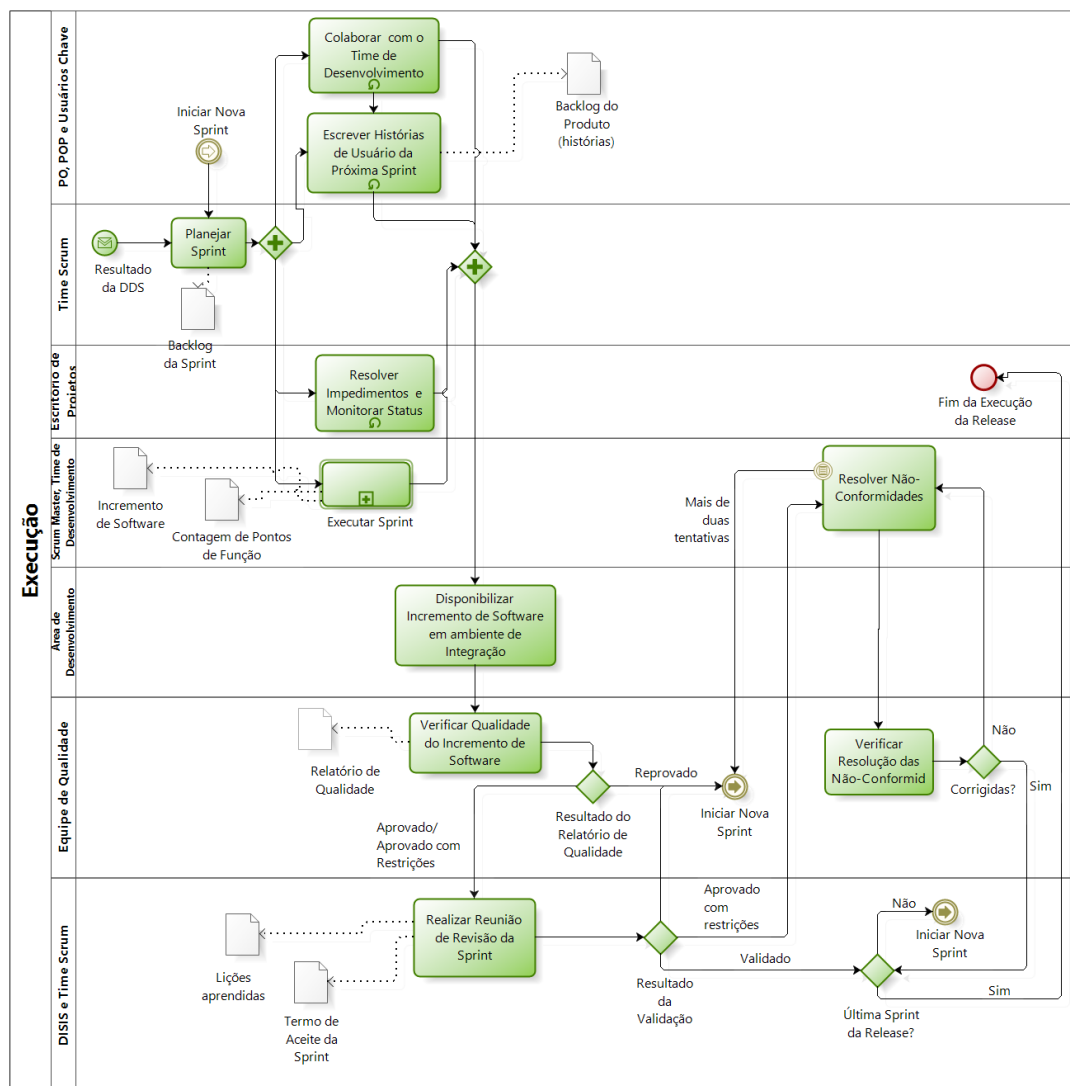
**Figura 25 – Subprocesso de Planejamento da Release no PAGDDS**

O subprocesso Execução (Figura 26 – Subprocesso de Execução no PAGDDS) incorpora em si os eventos do *Framework Scrum* definidos no Guia *Scrum* (SCHWABER; SUTHERLAND, 2013) a exceção dos seguintes:

Reunião de Retrospectiva da *Sprint* que é incorporada na Reunião de Revisão da *Sprint* através do padrão *Scrum* (SUTHERLAND; HARRISON; RIDDLE, 2014) “*Scrumming o Scrum*” inserindo no *Backlog* a melhoria mais importante definida pelo *Time Scrum* ao fim da Reunião de Revisão da *Sprint*.

Reunião Diária, pois trata-se de um evento entre o Time de Desenvolvimento, não alocado e que não pode ser gerenciado pelo órgão (BRASIL, 2003).

Sendo que o Planejamento da *Sprint* ocorre como previsto pelo Guia *Scrum*.



**Figura 26 – Subprocesso de Execução no PAGDDS**

As duas atividades do PO não possuem *time-box* definidos pois dependem de sua disponibilidade e são definidas na agenda do PO que é parte do Documento de Visão da Solução. Executa-se a *Sprint* em paralelo com as atividades do PO.

Ao fim da atividade “Executar *Sprint*”, que deve acabar com tempo suficiente para a execução das outras atividades do subprocesso Execução dentro do tempo estipulado para cada *Sprint*.

A atividade seguinte é “Disponibilizar Incremento de Software em ambiente de Integração”, proposta para responder de forma incremental ao recebimento provisório de

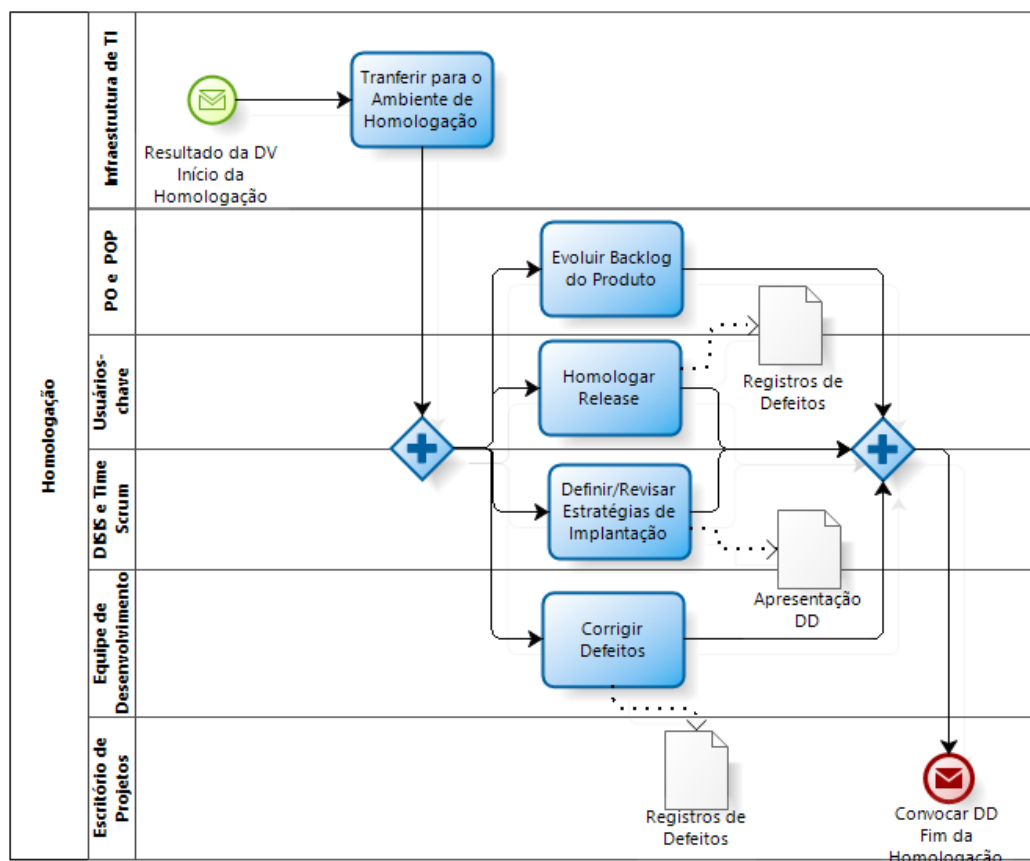


parte do produto/serviço contratado, eliminando assim o prazo de 15 dias estipulado em lei (BRASIL, 1993) para recebimento após comunicação de conclusão. Dessa forma encerra-se em um único ato tanto o comunicado de conclusão do incremento quanto o recebimento provisório deste.

Antes de ir para a reunião de Revisão da *Sprint* a qualidade técnica do Incremento de *Software* produzido é avaliada inclusive com a definição *Done* da *Release*, evitando que um produto em qualidade inferior a acordada em contrato passe para ser validado pelo cliente.

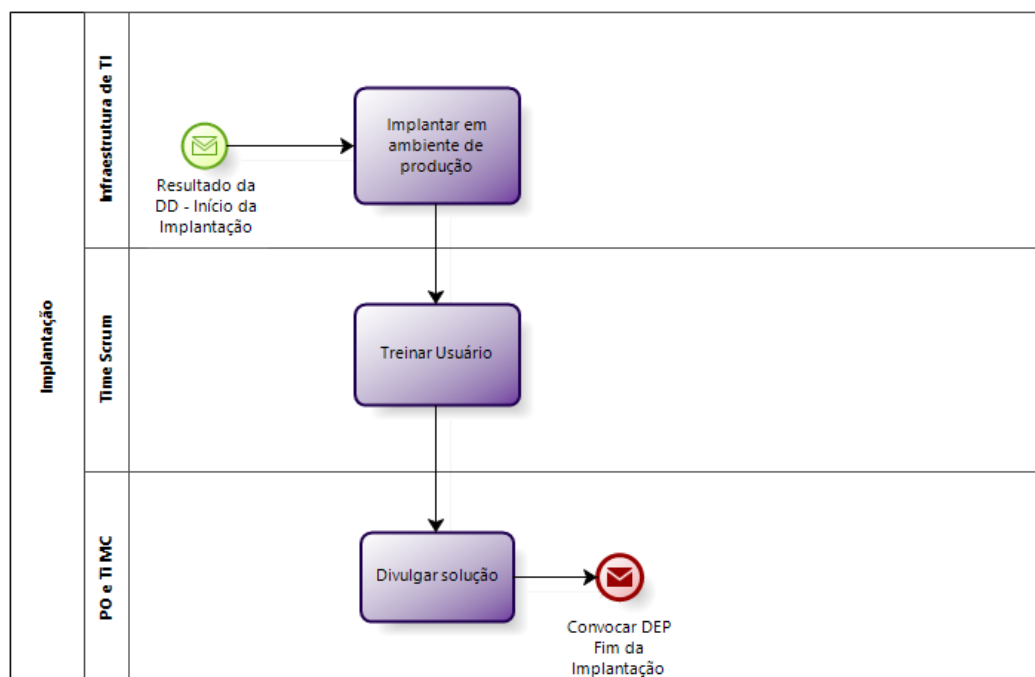
Da reunião de Revisão da *Sprint*, que segue a forma definida no Guia *Scrum* (SCHWABER; SUTHERLAND, 2013) resulta o Termo de Aceite da *Sprint* que posteriormente servirá de insumo para emissão do Termo de Recebimento Definitivo estipulado pela Lei nº 8.666, de 21 de junho de 1993 (BRASIL, 1993) em conjunto com os Relatórios de Qualidade.

O subprocesso de Homologação (Figura 27 - Subprocesso de Homologação do PAGDDS) não seria necessário pelo envolvimento tanto do PO quanto dos Stakeholders que estariam acompanhando o projeto (GALLARDO-VALENCIA; SIM, 2009), mas para este estudo de caso considerando o envolvimento não suficiente ou a necessidade específica de cada projeto este subprocesso se mantém. Sugere-se seu planejamento como sendo o de uma *Sprint* de validação (LEFFINGWELL, 2011).



**Figura 27 - Subprocesso de Homologação do PAGDDS**

A Implantação (Figura 28 - Subprocesso de Implantação do PAGDDS) deve ocorrer de acordo com o planejado pelo PO na “Estratégia de Desenvolvimento”. Não há cerimônias específicas para isto e sendo necessárias serão inseridas na atividade treinar usuários que vai depender da estratégia definida pelo Time *Scrum* que pode ir da liberação de manuais a treinamentos específicos.



**Figura 28 - Subprocesso de Implantação do PAGDDS**